

Т.Л. Партыка, И.И. Попов

[Операционные системы, среды и оболочки]



Т. Л. Партыка, И. И. Попов

ОПЕРАЦИОННЫЕ СИСТЕМЫ, СРЕДЫ И ОБОЛОЧКИ

5-е издание, переработанное и дополненное

*Допущено Министерством образования и науки Российской Федерации
в качестве учебного пособия для студентов учреждений среднего
профессионального образования, обучающихся по специальностям
информатики и вычислительной техники*

УДК 002.56(075.32)

ББК 32.973я723

П57

Рецензенты:

Б. В. Евтеев — доцент кафедры «Проектирование автоматизированных информационных систем» РЭА им. Г. В. Плеханова, кандидат физико-математических наук;

А. А. Емельянов — директор Института компьютерных технологий МЭСИ, заведующий кафедрой «Общая теория систем и системного анализа», доктор экономических наук, профессор;

В. П. Агальцов, В. А. Макунин — председатели предметных (цикловых) комиссий Математического колледжа

Партыка Т. Л.

П57 Операционные системы, среды и оболочки : учебное пособие / Т. Л. Партыка, И. И. Попов. — 5-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2017. — 560 с. : ил. — (Профессиональное образование).

ISBN 978-5-91134-743-7 (ФОРУМ)

ISBN 978-5-16-006733-9 (ИНФРА-М)

Рассматриваются общие принципы организации, состав, структура операционных систем и их оболочек, а также ряд конкретных систем. Значительное внимание уделяется проблемам управления информацией, процессами в ЭВМ и связи с оператором в рамках различных интерфейсов. В качестве примеров конкретных систем рассматриваются как ОС персональных компьютеров — MS DOS, Windows 3.x, 95/98/ME, NT/2000/XP/Vista/W7, Mac OS, так и ОС для многопользовательских ЭВМ — OS 360/370/375, RSX, Unix, Linux. Рассмотрен ряд оболочек, расширяющих возможности ОС ЭВМ как с текстовым, так и с графическим интерфейсом.

Для учащихся техникумов, колледжей, а также студентов вузов.

УДК 002.56(075.32)

ББК 32.973я723

ISBN 978-5-91134-743-7 (ФОРУМ)
ISBN 978-5-16-006733-9 (ИНФРА-М)

© Партыка Т. Л., Попов И. И.,
2003, 2007, 2010, 2012, 2014
© Издательство «ФОРУМ»,
2003, 2007, 2010, 2012, 2014

Предисловие

https://t.me/it_boooks

Функционирование современных ЭВМ обеспечивается на паритетных началах аппаратными и программными средствами. *Программное обеспечение* выполняет функцию посредника между пользователями и ЭВМ, расширяет возможности аппаратуры вычислительной машины, являясь логическим ее продолжением. Использование развитого программного обеспечения позволяет увеличить производительность вычислительных систем, автоматизировать многочисленные рутинные информационные процессы в различных областях человеческой деятельности, повысить производительность труда разработчиков различных систем автоматизированной переработки информации, сократить общие сроки разработок и т. д.

Программное обеспечение можно разделить на системное и прикладное. *Системное программное обеспечение* представляет собой комплекс управляющих и обрабатывающих программ, описаний и инструкций, обеспечивающих функционирование вычислительной системы, а также разработку и исполнение программ пользователей. Состав системного программного обеспечения практически не зависит от характера решаемых задач пользователей.

Программы системного программного обеспечения различаются по функциональному назначению и характеру исполнения. Они делятся на испытательные программы, системы программирования (СП) и операционные системы (ОС).

В состав системного программного обеспечения входят также сервисные программы, которые предназначены для проверки исправности блоков ЭВМ, обнаружения и локализации отказов устройств и устранения их влияния на работу системы в целом.

На базе операционных систем строятся программные средства, расширяющие функции ОС, и пакеты общего назначения для решения различных научных, технических, экономических и

других задач. Такие пакеты не входят в ОС и поставляются отдельно.

Прикладное программное обеспечение представляет собой совокупность программ решения конкретных задач из различных сфер применения ЭВМ. Специализированные комплексы программ решения конкретных задач вместе с сопровождающей документацией называют *пакетами прикладных программ* (ППП) или приложениями [1, 3].

Объем программного обеспечения современных вычислительных систем непрерывно возрастает, несмотря на то, что его стоимость остается довольно высокой даже при использовании промышленных методов разработки.

Особенно велика роль системного программного обеспечения, так как на его базе разрабатывается специальное программное обеспечение. Нередко доля стоимости системного программного обеспечения от общей стоимости вычислительной системы достигает 70 % и выше.

В исторической последовательности развития программных средств первыми появились узко ориентированные приложения («*программа, предназначенная для вычисления числа π с точностью до 200-го знака*», «*программа, предназначенная для расчета и печати платежной ведомости*» и пр.), затем СП (ранние их версии назывались *системами автоматизации программирования*) и ОС.

Ранние ЭВМ не предусматривали ОС, поэтому процедуры запуска/остановки программ, присоединения внешних носителей осуществлялись вручную или командами из прикладных программ. В середине 1960-х гг. ряд ведущих фирм-производителей ЭВМ, таких как IBM (International Business Machines — США), ICL (International Computer Limited — Великобритания), СІІ (Compagne Internationale pour Informatique — Франция), практически одновременно приступили к выпуску моделей машин (соответственно — IBM 360, System 4, Iris 80), оснащенных операционными системами (operating system).

Как показало время, наиболее совершенной и конкурентоспособной оказалась система OS/360 (IBM), в которой были заложены практически все основные черты ОС, позволяющие превратить ЭВМ в «автоматизированную фабрику» обработки информации при минимальном участии человека. OS/360 и другие современные ей системы были ориентированы на обработку потока заданий (или пакетную обработку — batch processing), при которой пользователь не мог вмешиваться в ход выполняемой за-

дачи, оперативно просмотреть промежуточные данные, т. е. был оторван от машины.

Появление и широкое распространение видеотерминалов создали возможность предоставить пользователю активный диалоговый доступ к вычислительному процессу, которым он занимается. В OS/360 фирмой и пользователями были внесены дополнения — появились системы TSO (Time Sharing Option — Система с разделением времени), CICS (Customer Information Control System — Система управления потреблением информации). Известен ряд удачных отечественных разработок — PRIMUS, FOCUS. Появившиеся в последующий период ОС ориентировались преимущественно на интерактивную работу пользователей — RSX, VMS, Unix и пр.

По-видимому, в настоящее время наиболее распространенной является интерактивная ОС Unix, версии которой разработаны практически для всех моделей ЭВМ. Для IBM PC-совместимых ЭВМ (ПЭВМ) в свое время была создана Unix-подобная система MS DOS (фирма MicroSoft). Следует согласиться с остроумным замечанием Питера Нортон о том, что «MS DOS — это Unix для дошкольников, Unix — это MS DOS для пользователей с высшим образованием».

ОС является первичной программной оболочкой для всякой ЭВМ — без ОС машина оказывается неодушевленным предметом. При включении электропитания ЭВМ автоматически осуществляется считывание с магнитного носителя, запись в оперативную память и запуск резидентных программ ОС или загрузка ОС (loading). В некоторых системах процесс загрузки прерывается для запроса у оператора адреса (номера), внешнего устройства, на котором размещаются программы ОС (резидентного устройства). При включении ПЭВМ поиск устройства с ОС осуществляется автоматически.

Резидентное устройство (как правило, НМД или CD) часто называют *bootable*, а процесс загрузки — *boot* («обувать»), что хорошо иллюстрирует, во-первых «голый» (или «необутый») статус компьютера без ОС, во-вторых — возможность «одеть» компьютер в разные ОС, при этом «образ машины» может измениться до неузнаваемости. Это давно и хорошо известно профессиональным пользователям больших ЭВМ, а сейчас стало и «достоянием широких масс» в связи с тем, что современные модели настольных ПК прекрасно работают с различными ОС — MS DOS, OS/2, Windows 95/98/ME/NT/2000/XP, Unix и пр.

ОС выполняет следующие основные (тесно взаимосвязанные) функции:

- управление данными;
- управление задачами;
- связь с внешней средой.

Структурно операционная система представляет собой совокупность программ, управляющих ходом работы вычислительной машины, идентифицирующих прикладные программы и данные и осуществляющих связь между машиной и оператором.

В настоящем учебном пособии рассматриваются общие принципы организации, состав, структура операционных систем и их оболочек, а также ряд конкретных систем.

В первой главе рассматриваются основные принципы организации и функционирования операционных систем, их состав и структура. Рассматриваются функции управления данными, включая планирование размещения данных и оперативное управление их прохождением через систему в процессе решения задач. С данной функцией тесно взаимосвязано управление заданиями (процессами, задачами), дисциплины обслуживания процессов и подпроцессов, управление очередями, оптимизация использования памяти для размещения задач. Функция связи с оператором реализует совокупность интерфейсов как предназначенных для текущего управления вычислительным процессом (оператор ЭВМ), так и для конфигурирования и установки ОС и ее компонент (системный администратор) и для передачи данных в прикладную программу и их вывода из нее (прикладной пользователь).

Во второй главе рассматриваются операционные системы персональных компьютеров (однопользовательские, однопрограммные) как получившие наибольшее распространение (сотни миллионов экземпляров). Это прежде всего операционная система MS DOS (на примере версии 6.22), затем графические программные оболочки Windows 3.x, операционные системы OS/2, Windows 95/98/ME, системы Windows NT/2000/XP/Vista/W7. Возможно, сюда следовало бы отнести также ОС Linux и версии Unix для ПЭВМ (AIX, Xenix), однако авторы поместили эти вопросы в следующую главу. Это связано, скорее, с историческими и генеалогическими соображениями — рассматриваемые во второй главе системы, во-первых, являются продукцией Microsoft (в основном), а, во-вторых, их возможности развивались вместе с аппаратной платформой, на которую они ориентировались, —

Intel — 8086-80286-386-486-Pentium и т. д. И хотя Windows NT/2000, как уверяют специалисты, ненамного уступает по сетевым и многопользовательским свойствам таким ОС, как Unix/Linux, следует не забывать, что Unix/Linux пришли на ПЭВМ с больших вычислительных систем (mainframes, hosts), а не наоборот.

В третьей главе рассматриваются многопользовательские многозадачные операционные системы в той исторической последовательности, в которой они были разработаны. Прежде всего, это ряд систем OS/360/370/375, классический прототип всех последующих разработок, затем операционные системы RSX (ОС РВ) и наиболее популярные сегодня среди системных администраторов мощных машин системы Unix и Linux. Основной чертой данных ОС является обеспечение работы систем в одном из следующих режимов:

- *системы с разделением времени*, в которых каждый участник «как бы монопольно» (виртуально) пользуется ресурсами ЭВМ, а основной задачей администраторов и разработчиков является защита данных от несанкционированного доступа и *взаимная изоляция* участников;
- *системы обеспечения групповых решений (СОГР)* — Computer Supported Cooperative Work, groupware — ориентированные на прямо противоположную задачу — обеспечить взаимодействие пользователей в процессе принятия решений. СОГР сочетают коммуникационную, вычислительную технологии и процедуры принятия решений для облегчения формулирования и решения неструктурированных проблем группой лиц.

В четвертой главе рассматриваются среды и оболочки операционных систем. Прежде всего, дается краткое описание программ расширения возможностей пакетных ОС (OS/360/370/375) — диалоговые мониторы ЕС ЭВМ, затем в исторической последовательности появления на сцене — монитор PCTOOLS для ПЭВМ, оболочка Norton Commander (NC), Norton Commander-подобные оболочки для Windows (в том числе NC для Windows, Windows Commander, Far Manager), программная оболочка Dosshell. Конечно, если строго придерживаться типизации оболочек на текстовые и графические, то средства NC для Windows и Windows Commander следовало бы рассматривать в одном ряду с Windows 3.x как графические, однако они помещены в 4-ю главу как прямые потомки NC, функционально сходные с Far Manager, который является принципиально тек-

стовой оболочкой. Здесь также рассмотрен Konqueror — функциональный аналог Fag для ОС Linux, а также ряд других приложений, расширяющих возможности пользователей ОС.

Учебное пособие базируется на материалах, накопленных авторами в процессе практической, исследовательской, а также преподавательской (МИФИ, РГГУ, РЭА им. Г. В. Плеханова) деятельности. Авторы выражают благодарность коллегам, принявшим участие в обсуждении материала: А. Г. Романенко (РГГУ), К. И. Курбакову (РЭА им. Г. В. Плеханова), П. Б. Храмцову (РНИЦ «Курчатовский институт»), рецензентам, а также студентам РГГУ и РЭА им. Г. В. Плеханова за предоставленные иллюстративные материалы.

Глава 1

ОПЕРАЦИОННЫЕ СИСТЕМЫ ЭВМ. ОСНОВНЫЕ ПРИНЦИПЫ И ПОНЯТИЯ

Операционная система — это совокупность программ, обеспечивающих организацию вычислительного процесса на ЭВМ. Основные задачи ОС следующие:

- увеличение пропускной способности ЭВМ (за счет организации непрерывной обработки потока задач с автоматическим переходом от одной задачи к другой и эффективного распределения ресурсов ЭВМ по нескольким задачам);
- уменьшение времени реакции системы на запросы пользователей пользователями ответов от ЭВМ;
- упрощение работы разработчиков программных средств и сотрудников обслуживающего персонала ЭВМ (за счет предоставления им значительного количества языков программирования и разнообразных сервисных программ).

1.1. Классы, функции и состав операционных систем

Функции ОС

Операционные системы могут классифицироваться по следующим показателям:

- *количеству пользователей*: однопользовательские ОС (MS DOS, Windows) и многопользовательские ОС (VM, Unix);
- *доступу*: пакетные (OS/360), интерактивные (Windows, Unix), системы реального времени (QNX, Neutrino, RSX);
- *количеству решаемых задач*: однозадачные ОС (MS DOS) и многозадачные ОС (Windows, Unix).

Операционная система предназначена для выполнения следующих основных (тесно взаимосвязанных) функций (рис. 1.1):

- управление данными;
- управление задачами (заданиями, процессами);
- связь с человеком-оператором.



Рис. 1.1. Основные функции операционных систем (ОС)

В различных ОС эти функции реализуются в различных масштабах и с помощью разных технических, программных, информационных методов и средств.

Системное программное обеспечение ЭВМ предназначено для осуществления адаптируемости программ пользователей к изменениям состава ресурсов ЭВМ. Высокая производительность вычислительной системы обеспечивается операционной системой благодаря применению мультипрограммирования и наличию специальных программных средств для выполнения трудоемких операций ввода-вывода информации. Высокая производительность труда программиста достигается за счет предоставления ему большого числа языков программирования; специальных библиотек программ; удобных средств ввода-вывода, средств отладки программ и оформления заданий.

К числу наиболее известных первых управляющих программ относятся комплексы SAGE, SABRE, MERCURY, реализованные на ЭВМ второго поколения. Для ЭВМ IBM/360 были разработаны операционные системы, обеспечивающие пакетную технологию обработки данных и работу в реальном масштабе време-

ни, а также реализацию многомашинных и мультипроцессорных комплексов.

Первая функционально полная ОС (OS/360) была предложена фирмой IBM в качестве оболочки ЭВМ IBM/360. Разработка и внедрение ОС позволили разграничить функции операторов, администраторов, программистов, пользователей, а также существенно (в десятки и сотни раз) повысить производительность ЭВМ и степень загрузки технических средств.

Ранние версии OS/360 были ориентированы на пакетную (batch processing) обработку информации — входной поток заданий (на МЛ, МД или перфокартах) подготавливался заранее и поступал на обработку в непрерывном режиме. В дальнейшем возникли расширения OS/360/375, допускающие диалоговую обработку данных с терминалов пользователя, последняя из версий (OS SVM) фактически предоставляла в распоряжение пользователя «виртуальную персональную ЭВМ» с полной мощностью вычислительной установки IBM/360/375. Операционные системы других семейств (поколений), например RSX (для PDP/11 DEC) или Unix, с самого возникновения ориентировались на интерактивное взаимодействие с пользователями.

Одно из основных требований к разработке программного обеспечения ЭВМ — *модульность*. Модульная структура программ и программных комплексов облегчает организацию работы больших коллективов программистов по созданию программного обеспечения. Другое важное требование к программному обеспечению — *возможность развития* программной системы. Выполнению этого требования способствует модульная организация программ. Существенным является требование *простоты* освоения, поддержания, эксплуатации и совершенствования возможностей программного обеспечения. Это позволяет обходиться небольшим числом специалистов, обслуживающих принятое к эксплуатации программное обеспечение.

Система программного обеспечения предназначена для эксплуатации многочисленными группами пользователей в различных организациях и предприятиях, поэтому она должна обладать свойствами *гибкости, адаптируемости*. Эти требования обеспечиваются применением принципов открытости, машинной независимости обрабатываемых программ, унификации использования периферийного оборудования и т. д. По возможности должна достигаться *совместимость* программного обеспечения различных ЭВМ и систем обработки данных. Как правило, совмести-

мость программ обеспечивается в рамках ЭВМ одной аппаратной платформы. Программная совместимость для различных семейств ЭВМ достигается на уровне языков программирования.

Требование *минимальности вмешательства человека* в процесс обработки информации на ЭВМ удовлетворяется путем автоматизации различных этапов вычислительного процесса. В частности, автоматическое распределение ресурсов повышает эффективность использования вычислительной системы. Программное обеспечение должно удовлетворять также требованиям *параметрической универсальности*, *функциональной избыточности* (наличия в системе нескольких программ, реализующих одну и ту же функцию), *функциональной избирательности* (возможность конфигурирования программного обеспечения в соответствии с потребностями и возможностями конкретного пользователя).

Техническая документация на программные средства, являющаяся одним из важнейших элементов программного обеспечения, должна оформляться по единым стандартам. К технической документации относятся графические и текстовые документы, определяющие назначение, состав и структуру созданного программного средства. В них должны содержаться сведения, необходимые для тестирования, приемки, обучения пользователей, эксплуатации и наращивания возможностей программ. Выпуск документации является трудоемким процессом, поэтому желательно его автоматизировать. Документация на программное обеспечение должна удовлетворять требованиям *единства терминологии*, номенклатуры и наименования документов, единой системы обозначений в документах, идентичности документов независимо от места их разработки. Кроме того, должны соблюдаться единые правила внесения изменений, учета и хранения документации. Детальность описания отдельных модулей программного обеспечения должна соответствовать уровню подготовки потенциальных пользователей (системного программиста, программиста-пользователя, оператора и т. д.).

Программы ОС

Основные программы ОС постоянно (резидентно) занимают в оперативной памяти (ОП) объем, установленный при конфигурировании системы. Остальные части операционной системы по мере необходимости вызываются из внешней памяти на МД.

Операционная система обеспечивает осуществление в вычислительной системе следующих процессов:

- обработки задач;
- работы системы в режиме диалога и квантования времени;
- работы системы в реальном масштабе времени в составе многопроцессорных и многомашинных комплексов;
- связи оператора с системой;
- протоколирования хода выполнения вычислительных работ;
- обработки данных, поступающих по каналам связи;
- функционирования устройств ввода-вывода;
- использования широкого набора средств отладки и тестирования программ;
- планирования прохождения задач в соответствии с их приоритетами;
- ведения учета и контроля за использованием данных, программ и ресурсов ЭВМ.

Основные компоненты операционных систем — управляющие и обрабатывающие программы. Управляющие программы управляют работой вычислительной системы, обеспечивая в первую очередь автоматическую смену заданий для поддержания непрерывного режима работы ЭВМ при переходе от одной программы к другой без вмешательства оператора.

Управляющая программа определяет порядок выполнения обрабатывающих программ и обеспечивает необходимым набором услуг для их выполнения. Основные функции управляющей программы: последовательное или приоритетное выполнение каждой работы (управление задачами); хранение, поиск и обслуживание данных независимо от их организации и способа хранения (управление данными).

Программы управления задачами считают входные потоки задач, обрабатывают их в зависимости от приоритета, инициируют одновременное выполнение нескольких заданий; вызывают процедуры; ведут системный журнал.

Программы управления данными обеспечивают способы организации, идентификации, хранения, каталогизации и выборки обрабатываемых данных. Эти программы управляют вводом-выводом данных с различной организацией, объединением записей в блоки и разделением блоков на записи, обработкой меток томов и наборов данных.

Программы управления восстановлением после сбоя обрабатывают прерывания от систем контроля, регистрируют сбой в процессоре и внешних устройствах, формируют записи о сбое в журнале, анализируют возможность завершения затронутой сбоем задачи и переводят систему в состояние ожидания, если завершение задачи невозможно.

Конфигурация системы. Прикладная программа в операционных системах может получить от ОС в процессе своей работы характеристики конкретной реализации системы, в среде которой она функционирует: имя, версию и редакцию операционной системы, тип и технические характеристики компьютера. В ОС обычно имеются средства локализации, позволяющие настроить систему на конкретное национальное (местное) представление данных: представление десятичных дробей, денежных величин, даты и времени.

Операционные системы предоставляют программе пользователя возможность узнать текущие дату и время. За начало отсчета, например, в MS DOS, принята дата 1 января 1980 г. 0 часов 0 минут 0 секунд по Гринвичу, в Unix — 1 января 1970 г. Системы предоставляют возможность измерения временных интервалов короче 1 секунды с помощью специальных системных вызовов. ОС может переводить дату и время из внутреннего числового представления в символьное (пригодное к выводу, например, на терминал); местное время — во время по Гринвичу и наоборот; предоставлять информацию о часовом поясе, летнем и зимнем времени.

1.2. Управление данными в операционных системах

Управление данными включает следующие компоненты:

- долговременное планирование — организацию размещения данных на внешних носителях, их выборку и предоставление пользовательским программам;
- оперативное управление — распределение оперативной памяти под программы и данные, реализацию обмена данными между оперативной и внешней памятью;
- управление внешними устройствами ввода-вывода и размещения данных.

Внешние устройства ЭВМ

Несколько слов о тех устройствах, которые отвечают за размещение и ввод-вывод данных. Здесь мы ограничимся рассмотрением спектра устройств IBM PC-совместимых ПЭВМ. Прежде всего, необходимо отметить, что типовая конфигурация внешних устройств (ВУ) в данном случае включает: *терминал/консоль (экран и клавиатура), накопители на магнитных дисках (НМД) и принтер*. Эти устройства будут подробно рассмотрены ниже. Пока же вкратце охарактеризуем принципы функционирования ВУ и их перечень в целом. Прежде всего, контроллеры ВУ ПЭВМ представляют собой стандартного формата электронные платы (*интерфейсные карты, адаптеры* и пр.), которые практически полностью взаимозаменяемы, что позволяет укомплектовать экземпляр ПЭВМ любым желаемым набором устройств (но не более 4—8, в зависимости от класса машины).

Контроллеры типовых устройств, как правило, являются несъемными и размещаются на системной плате (motherboard) ПЭВМ. И далее, практически все устройства требуют для своей работы программной поддержки (как минимум — запуска и непрерывного функционирования специальных программ — *драйверов устройств*, или более сложных *прикладных программ*). В более мощных системах (Unix, Windows NT) подобные программы входят в состав *операционной системы* и обязательно присутствуют в машине. В случае же, например, MS DOS — это необязательные компоненты, которые необходимо приобретать и устанавливать дополнительно.

Накопители на магнитных носителях, файлы, циклы обработки

Накопители данного типа являются основной средой хранения информации в ЭВМ и разделяются на накопители на магнитных лентах (НМЛ) и магнитных дисках (НМД). Появившиеся в различное время магнитные барабаны и магнитные карты особого распространения не получили. В различные временные периоды НМД и НМЛ по очереди доминировали в качестве основного вида накопителя. В настоящее время устоялось следующее представление: НМД используются для оперативного (во время реше-

ния задач) хранения информации, НМЛ — для резервного (архивного) хранения (стримеры).

В состав внешних запоминающих устройств ЭВМ входят накопители на магнитных лентах (НМЛ), магнитных дисках (НМД), кассетных магнитных лентах (НКМЛ), дискетах или гибких магнитных дисках (НГМД) и т. д. Наибольшее распространение получили разработанные ранее других накопители на магнитных лентах и магнитных дисках.

Файл (набор данных на внешнем носителе) рассматривается как совокупность записей одинаковой структуры (обычно, хотя и необязательно — фиксированной длины), каждая из которых представляет собой набор (агрегат) разнородных данных (в языках программирования PL/1, Pascal, Си за подобными объектами так и закрепилось название *структура* — *structure*).

Понятие «файл» появилось впервые в операционной системе OS/360 фирмы IBM, причем в ранних версиях системы «настоящим файлом» считался только перфокарточный массив (file=картотека), данные на МД и МЛ обозначались как DS (Data Set — набор данных). В последующих ОС (RSX, Unix, MS DOS) файлами становятся именованные организованные наборы данных на любых носителях и устройствах, за сохранность и обновляемость которых (а также передачу в прикладные программы/из прикладных программ) и несет ответственность ОС ЭВМ.

Цикл обработки файла (например, внесение изменений в счета клиентов) включает следующие операции (рис. 1.2):

- открытие файла — занятие устройства, на котором файл размещен (например, МД), создание в оперативной памяти (ОП) *управляющего блока*, в котором записывается справка о состоянии файла, и *буфера* (или набора буферов — буферного пула) для хранения текущей, обрабатываемой записи файла;
- организацию цикла, управляемого файлом (заканчивается по исчерпанию записей файла — наступлении состояния EOF — end-of-file), после чего выполняется некоторый оператор (обычно освобождение устройства). Цикл должен содержать команду типа READ, GET (ввод записи) или PUT, WRITE (вывод записи) либо REWRITE (обновить запись). Команда READ может являться функциональным аналогом заголовка цикла;
- закрытие файла — выполнение операций по внесению всех окончательных изменений в файл и его реквизиты, освобо-

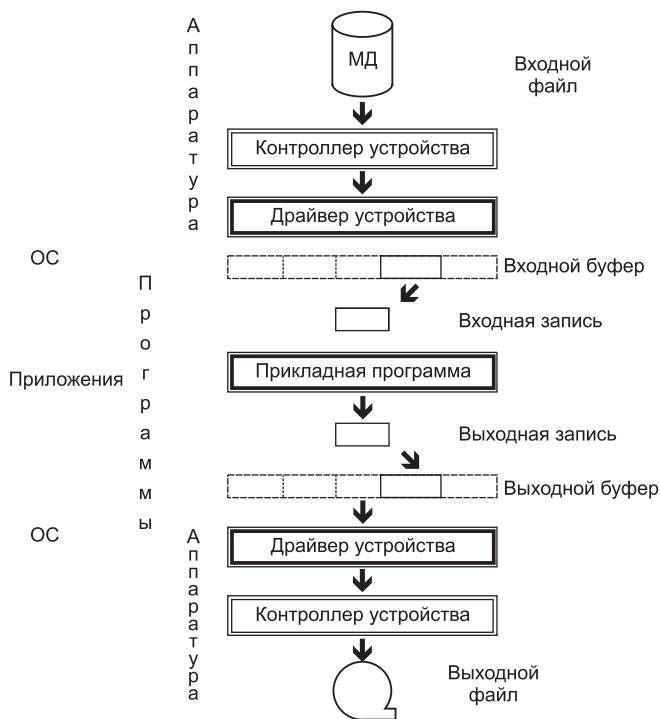


Рис. 1.2. Типичная траектория данных при обработке в ЭВМ

ждение памяти, отведенной под файл, и устройства, на котором он размещался.

Таким образом, траектория данных, обрабатываемых в компьютере, выглядит следующим образом:

- считывание (ввод) порции (*блок*) данных с *накопителя* (ВУ) и помещение его в область ОП (*буфер*);
- извлечение данных из буфера, их обработка и помещение обратно или в другой (выходной) буфер;
- после окончания обработки — вывод (запись) результатов на выходной накопитель также в форме одного или нескольких блоков.

То есть всякая правильно выполненная и завершенная операция обработки данных начинается считыванием информации с ВУ и должна заканчиваться записью результата на ВУ. Всякое прерывание данной последовательности неминуемо приводит к разрушению информации и потере данных.

Адресация, имена, спецификация данных в ОС

Управление данными. Понятие «управление данными» является характерным не только для ОС, но и для СУБД (систем управления базами данных). В чем же заключается различие?

На уровне ОС осуществляется связь между *адресом* данных и *именем* (файла). В эпоху до появления ОС и систем программирования (СП) программист должен был писать программы в непосредственных адресах ЭВМ. Элементом такой программы является команда в абсолютных адресах, например, как это было в очень популярной в свое время двухадресной машине Минск 22/32:

```
10 00 1234 7653
```

(«сложить содержимое адреса 1234_8 с содержимым адреса 7653_8 и записать по адресу 7653_8 »).

При этом управление данными на внешних носителях состояло в написании команд вида:

```
45 00 1200 0000
```

```
47 00 0002 1234
```

(«на устройстве накопления данных на МЛ перемотать ленту на 12_8 зон (блоков), затем прочитать 2 зоны в оперативную память, размещая данные с адреса 1234_8 »).

Операционные системы избавляют программиста от таких забот, предоставляя возможность оперировать файлами и их именами.

При этом в различных ОС приняты различные принципы именования данных.

1. Для OS/360 — «прабабушки» современных ОС — обозначение/описание файла могло бы выглядеть следующим образом:

```
UNIT=SYSDA, VOL=SER=MASTER, DSN=SYS2.PGMLIB(COPIER3),
```

что означает набор данных (файл), который размещен на устройстве прямого доступа (НМД — SYSDA), причем пакет дисков имеет имя (метку) MASTER, имя файла состоит из группового обозначения (SYS2) и индивидуального (PGMLIB), причем последнее соответствует партиционному (состоящему из разделов) файлу, раздел которого COPIER3 и является основным фигурантом данного описания.

2. Для ОС RSX 11/20 (предшественница Unix) обозначения выглядят так:

устройство: [g, n] имя.расширение; версия

где устройство — идентификатор устройства;

[g, n] — каталог (UIC — User Identification Code), идентификатор пользователя, состоящий из имени (номера) группы (g) и имени пользователя в группе (n);

имя — выбираемое пользователем наименование НД (не более 8 символов);

расширение — идентификатор файла (не более 3 символов), используемый для группирования файлов в типы.

Некоторые стандартные типы файлов, используемые в ОС и в пользовательских программах:

- .ftn — текст программы на Фортране;
- .bas — текст программы на ЯП Бейсик;
- .cmd — командный файл;
- .tsk — исполнительный модуль;
- .txt — текстовый файл.

Пример: DP0: [1, 7] ADABAS.TSK; 1 — это программный модуль с именем ADABAS, размещенный в директории [1, 7] на МД с адресом DP0:.

Сокращенное наименование файла может состоять только из имени. При этом устройство и [g, n] берутся из системных умолчаний или пользовательских назначений; расширение — задается стандартным типом файла; версия — максимальная из существующих.

Спецификация файлов — соглашения о кратком групповом обозначении некоторой совокупности обрабатываемых, переименовываемых, удаляемых, копируемых и пр. файлов.

В спецификации файлов могут использоваться символы маскирования «*» и «?», вносимые в компоненты обозначения файла, причем «*» соответствует допустимой строке символов, а «?» — одному допустимому символу.

Примеры:

[*, *]*.TSK; 2 — все файлы задач во всех директориях, 2-й версии;

[1, 5] ADA*.SYS — файлы директории [1, 5] с именем, начинающимся с ADA, расширением SYS, 1-й версии;

[5, 5] SYSTEM.?? — файлы с именем SYSTEM, имеющие 2-символьные расширения.

3. Unix-спецификация файла может иметь вид:

ROOT/USR/CPP/EXPERT2.C

Здесь описан полный путь для поиска файла (EXPERT2.C — текст программы на ЯП Си), включающий каталоги и подкаталоги ROOT, USR, CPP. Методы групповой спецификации файлов в ОС Unix подробно описаны ниже в соответствующем разделе.

4. MS DOS-спецификация файла в типовом случае выглядит так:

C:\WIN98_SE\PROGRAMMS\COMMAND.COM.

Методы групповой спецификации аналогичны указанным выше для RSX.

Вначале для большинства ОС были установлены ограничения на длину и состав имени файла, во многом аналогичные ограничениям на идентификаторы переменных, принятых в то время в языках программирования:

- имя может содержать только символы заглавной латиницы, цифры и подчеркивание;
- имя должно начинаться с буквы;
- длина имени файла не более 8 символов, длина расширения (типа) не более 3.

В дальнейшем, по мере развития и распространения ОС, эти ограничения во многом стали сниматься:

- появилось понятие «длинного имени файла», включающего ранее запрещенные символы (пробелы и пр.);
- были разрешены национальные символы в наименованиях файлов (кириллица и пр.).

Уровни доступа к данным, реализуемые ОС (либо ОС совместно с прикладными программами), приведены на рис. 1.3. При этом участок «Том — Каталог — Файл» реализуется во всех ОС. Участок «Блок — Строка — Слово — Символ» может поддерживаться как функциями ОС, так и в рамках прикладных программ. Иногда это распределение функций устанавливается пользователем (программистом) путем указания типа файла — например *записе-ориентированный* или *поток-ориентированный*.



Рис. 1.3. Уровни доступа к данным в абстрактной ОС

Вернемся к проблеме соответствия ОС и СУБД (с точки зрения управления данными). Надо отметить, что СУБД оперируют данными на содержательном уровне, хотя физические структуры, используемые для этих целей, могут быть аналогичны структурам, создаваемым ОС (см. ниже файловые системы ОС).

Коренное отличие СУБД от файловых систем ОС состоит в том, что СУБД устанавливает связь между *содержанием* и *адресом*, а ОС — между *именем* и *адресом* данных. В то же время эта грань постоянно подвергается «атакам» с обеих сторон.

Известен ряд ОС, перешедших эту грань (например, ОС/360 с «индексным доступом к данным», IN-PICK, включающая язык поиска записей файлов по содержанию, Unix, включающая команды сортировки, коррекции или объединения содержимого текстовых файлов, наподобие того, как это осуществляется с таблицами данных в СУБД).

Тем не менее следует признать это, скорее, исключением, чем правилом, и в компетенцию ОС надо относить только связь «имя — адрес», оставляя другие зависимости на ответственность прикладных программ и оболочек СУБД и АИПС (автоматизированные информационно-поисковые системы).

Накопители на магнитных лентах

Эти накопители относятся к классу внешних запоминающих устройств последовательного доступа. В них доступ к требуемому набору данных происходит только после завершения перемотки всей предшествующей части магнитной ленты (МЛ). Такие накопители благодаря низкой стоимости, простоте эксплуатации и хранения, компактности и долговременности использования обладают несомненными преимуществами в тех случаях, когда порции данных обрабатываются последовательно друг за другом.

Магнитные ленты для цифровой записи данных размещаются на бобинах или кассетах (подобно лентам для бытовой аудио-или видеозаписи). Однако принципы размещения информации на МЛ в данном случае существенно другие — рис. 1.4:

- информация размещается на носителе в виде блоков (массивов данных фиксированной или переменной длины);
- информационные блоки разделены пустыми промежутками (*gap*), позволяющими считывающему устройству распознать начало (окончание) блока. Размер промежутка между



Рис. 1.4. Структура данных на магнитных лентах:

1 — физическое начало ленты (начальный ракорд); 2 — информационные блоки (ИБ) 1-го файла; 3 — GАР, промежуток между блоками; 4 — EOF — end-of-file — служебный блок, задающий конец 1-го файла; 5 — информационные блоки 2-го файла; 6 — конец 2-го файла; 7 — EO (end-of-volume) — служебный блок, задающий логический конец ленты; 8 — физический конец ленты (ракорд)

записями выбирается достаточным для разгона ленты до установленной скорости и остановки ее точно на следующем промежутке. Недостаток использования промежутков между записями — уменьшение полезного объема МЛ, так как области, отведенные под промежутки, нельзя использовать для хранения данных. Частично указанный недостаток устраняет *процесс блокирования*, суть которого состоит в объединении нескольких записей в блоки;

- блоки разделяются на *информационные* (ИБ — распознаются программами) и *служебные* (распознаются устройством — конец файла и конец тома);
- физическое начало и физический конец ленты обычно определяются оптическим или механическим образом (независимо от содержания ленты).

В ЭВМ обычно применяется девятидорожечная магнитная лента. Информация записывается одновременно девятью магнитными головками. Из девяти одновременно записываемых битов информации восемь являются информационными (один байт) и один — контрольным битом четности. Начало области магнитной ленты, в которую записывается информация, называется точкой загрузки и помечается специальным физическим маркером. Физический маркер представляет собой кусочек алюминиевой фольги, наклеиваемый на расстоянии от начала магнитной ленты. Конец информационной области МЛ помечается таким же физическим маркером, наклеиваемым на расстоянии от конца МЛ. Наличие указанных специальных маркеров, распознавание которых производится фотоэлектронным способом, позволяет осуществить перемотку МЛ к началу информационной области и автоматический останов по достижении ее конца.

Максимальное ограничение на размер блока зависит от размера доступной оперативной памяти (возможность размещения буфера считывания файла). Блокирование увеличивает полезный

объем магнитной ленты за счет сокращения числа промежутков между записями. Кроме того, уменьшается количество операций ввода-вывода, так как за одну операцию производится пересылка не одной записи, а сразу нескольких. Преимущества блокирования, заключающиеся в увеличении полезного объема МЛ и уменьшении общего времени работы программы на ввод-вывод данных, значительно превосходят возникающие при этом недостатки, связанные с увеличением объемов данных в программе пользователя и необходимостью выполнения процедур по формированию блоков и разделению принятых блоков на записи.

Устройство записи-считывания информации с МЛ («магнитофон») позволяет осуществлять следующие операции:

- пропустить (вперед или назад) N ИБ;
- пропустить (вперед или назад) N файлов;
- прочитав (записать) блок;
- прочитав (записать) файл;
- позиционировать на конец файла (для записи дополнительных ИБ в этот файл; очевидно, что данные последующих файлов будут затерты);
- позиционировать на начало ленты;
- позиционировать на логический конец ленты (для записи дополнительного файла).

Очевидно, если блок EOV будет записан в начале ленты, то все файлы становятся логически недоступными, и лента приобретает статус *инициализированной*.

Значение контрольного бита четности выбирается в зависимости от значений восьми информационных битов. Если число единиц в восьми информационных битах нечетное, то в контрольный бит четности заносится единица, а если четное — нуль. Таким образом, общее число единиц в девяти записываемых битах всегда должно быть четным, это контролируется в процессе чтения данных.

Индикатором возникшей ошибки является нечетное число битов в считываемом с МЛ символе. Причинами ошибок часто бывают дефекты покрытия МЛ и налипание на ее поверхности пыли.

Предусмотрена возможность пропуска выявленных дефектных участков на МЛ. Помимо посимвольного контроля, производимого с помощью контрольного бита четности, существует поблочный контроль данных. Его суть заключается в том, что в конце каждого блока записывается контрольная комбинация.

В случае возникновения в блоке единичной ошибки посимвольный и поблочный контроль позволяет определить ее местонахождение и выполнить автоматическое исправление. Для этого перед байтом поблочного контроля записывается байт циклического контроля. После обнаружения ошибки делается предположение о наличии временного дефекта МЛ и осуществляется повторная попытка записи информации на то же место. Если и последующие попытки заканчиваются неудачей, то дефектный участок пропускается. В целях контроля правильности выполнения операций записи-чтения помимо основного набора магнитных головок используется дополнительный.

С помощью дополнительного набора магнитных головок считываются только что записанные на МЛ биты информации, в случае их несовпадения идентифицируется состояние ошибки. Оба набора магнитных головок считывают данные с МЛ и при несовпадении какой-либо пары битов также будет выработан сигнал об ошибке.

Магнитные ленты в силу ряда своих положительных достоинств играют важную роль при организации больших информационных архивов и фондов пакетов программ.

Размещение информации на МЛ связано со следующими проблемами. Для уверенного распознавания промежутка (gap) он должен иметь значительную длину (особенно при высоких скоростях перемотки/чтения). При скорости движения ленты 2—3 м/с длина промежутка должна составлять не менее 1—2 см. Очевидно, что для того чтобы эффективность использования МЛ была достаточно высокой, длина ИБ должна как минимум в 2—3 раза превышать длину промежутка (при этом коэффициент полезного использования МЛ будет составлять 60—75 %). При этом также увеличивается скорость обмена между ОП и ВУ, так как за одно обращение к МЛ считывается как минимум один ИБ. Однако увеличение длины ИБ требует увеличения объема ОП для размещения буфера, связанного с данным файлом (буфер выделяется операционной системой при открытии файла), в связи с чем одновременное открытие большого числа файлов может оказаться невозможным при ограниченном размере ОП.

В накопителях на магнитной ленте кассетного типа (НКМЛ) носителем информации обычно является стандартная компакт-кассета с магнитной лентой шириной 3,81 мм и длиной 90 м. По сравнению с бобинными магнитными лентами преимущества НКМЛ состоят в компактности и менее высокой стоимо-

сти. Их недостатки заключаются в меньшей скорости обмена данными и в десятки раз меньшей информационной емкости.

Основной недостаток внешних запоминающих устройств на магнитных лентах — значительное время ожидания на помещение требуемой области магнитной ленты в зону магнитных головок для выполнения операции записи (считывания). Это занимает в большинстве случаев несколько десятков секунд, что существенно замедляет процесс обработки данных. Прогресс в этой области был достигнут путем разработки запоминающих устройств прямого доступа, включающих в свой состав накопители на магнитных барабанах и дисках (НМД), на гибком магнитном диске (НГМД) и кассетном магнитном диске (НКМД).

Накопители на магнитных дисках

Накопители на магнитных дисках получили наибольшее распространение. В них каждая запись данных имеет свой собственный уникальный адрес, обеспечивающий непосредственный (минуя все остальные записи) доступ к ней. В НМД предусмотрена аналогичная НМЛ возможность последовательного доступа к информации. Конструкция НМД сложнее, чем у НМЛ, а следовательно, выше их стоимость. В НМД в качестве носителей данных используется *пакет магнитных дисков*, закрепленных на одном стержне, вокруг которого они вращаются с постоянной скоростью. Поверхность магнитного диска, покрытая ферромагнитным слоем, называется *рабочей*.

Каждый магнитный диск пакета, кроме верхнего и нижнего, имеет две рабочие поверхности. Верхний и нижний магнитные диски обладают по одной *рабочей поверхности*, расположенной соответственно на нижней и верхней частях указанных дисков. Каждая рабочая поверхность магнитного диска разбита на N окружностей (*дорожек*), пронумерованных от 0 до $N-1$ от края к центру. На каждой из дорожек начало области данных механически идентифицировано с помощью маркера начала оборота. Дорожки, расположенные одна под другой на разных магнитных дисках, образуют соответственно N *цилиндров*.

Из N цилиндров M являются резервными и $N - M$ — основными. Дорожки резервных цилиндров пользователям недоступны. Системные средства обеспечивают замену дорожки основного цилиндра, ставшей дефектной, на дорожку запасного цилиндра.

Запись и считывание информации в НМД производит механизм доступа, состоящий из держателей магнитных головок (блок магнитных головок).

Количество магнитных головок равно числу рабочих поверхностей на одном пакете дисков. Если пакет состоит из 11 дисков, то механизм доступа состоит из 10 держателей с двумя магнитными головками на каждом из них. Держатели магнитных головок объединены в единый блок таким образом, чтобы обеспечить их синхронное перемещение вдоль всех цилиндров. Фиксируя блок механизма доступа на каком-либо из цилиндров с помощью только электронного переключения головок, можно сделать переход с одной дорожки на другую данного цилиндра. При фиксированном положении блока механизма доступа возможно обращение к любой из записей, находящихся на дорожках текущего цилиндра. Дорожки в цилиндре нумеруются начиная с верхних. Как правило, обращение к дорожкам происходит с нулевой по последнюю одного цилиндра, потом с нулевой дорожки следующего цилиндра и т. д.

Любая операция чтения (записи) информации с (на) магнитного диска состоит из трех этапов. На первом этапе происходит механический подвод магнитной головки к дорожке, содержащей требуемые данные. На втором этапе обеспечивается ожидание момента, пока требуемая запись не окажется в зоне магнитной головки. На третьем этапе осуществляется собственно процесс обмена информацией между вычислительной машиной и магнитным диском. Таким образом, общее время, затрачиваемое на операцию записи-считывания, состоит из суммы времен поиска соответствующей дорожки, ожидания подвода записи (так называемое время ротационного запаздывания) и обмена с ЭВМ. Максимальное значение времени ротационного запаздывания равно времени, за которое совершается полный оборот магнитного диска.

В идейном плане размещение информации на МД аналогично МЛ (дорожка МД эквивалентна отрезку МЛ). Адрес информационного блока на МД состоит из номера дорожки и номера блока на дорожке. Начало и конец блока распознаются по промежуткам между блоками, начало и конец дорожки — оптическим (для сменных МД) или электромагнитным (для постоянных МД) датчиком угла поворота оси пакета МД.

Размер блока, очевидно, не может быть больше длины дорожки МД. Считывающее устройство в данном случае ориентировано на выполнение единственной операции — прочитать

(или записать) информационный блок, который задан своим адресом. За считывание файла несет ответственность операционная система, поддерживающая файловые структуры на МД.

Соображения по поводу длины блоков, отмеченные выше по поводу МЛ, сохраняют свою силу и для МД, однако здесь возникают и некоторые дополнительные сложности. Использование блоков фиксированной длины на МЛ не дает никаких преимуществ, в то время как для НМД использование блоков фиксированной длины позволяет использовать датчик угла поворота как дополнительный идентификатор конца блока, что приводит к увеличению КПД использования поверхности диска.

При создании нового файла операционная система выделяет под его размещение по меньшей мере один блок, и если в среднем длина файла оказывается намного меньше размера блока, коэффициент использования МД оказывается низким. Тем самым, если предполагается обрабатывать большое число файлов малого объема, то целесообразно задать небольшую стандартную длину блока. Таким образом, выбор длины блока данных на МД определяется противоречивыми факторами как *за* увеличение длины, так и *против* этого.

Особенности и характеристики НМД для персональных компьютеров

Различают магнитные диски: жесткие (НЖМД, HDD, «винчестер») и гибкие (НГМД, FDD, «флоппи»). HDD являются более скоростными устройствами, чем FDD.

Винчестер (HDD) — накопитель на несъемном пакете магнитных дисков, был создан в 1973 г. Все магнитные диски (объединенные в *пакет дисков*) герметически «упакованы» в общий кожух. Магнитные диски не могут изыматься из HDD и заменяться на аналогичные.

Флоппи (FDD) (первоначально — разработка фирмы IBM) — накопитель на съемном гибком магнитном диске (флоппи). Флоппи-диск имеет пластиковую основу и находится в пластиковом кожухе. Флоппи-диск вставляется в FDD вместе с кожухом и вращается внутри кожуха со скоростью 300 об/мин.

В персональных компьютерах используются два типоразмера FDD: 5.25" (дискета заключена в гибкий пластиковый кожух) и 3.5" (дискета 3.5" заключена в жесткий пластиковый кожух).

Магнитная поверхность разбивается на дорожки (концентрические окружности, рис. 1.5). Дорожки нумеруются начиная с 0-й (максимальный радиус). Магнитная поверхность «разбита» также на секторы. Секторы нумеруются начиная с 1-го. Размер каждого сектора обычно равен 512 байт (для MS DOS). Физический адрес сектора составляется как сумма (точнее, конкатенация) соответствующих номеров: № поверхности, № дорожки, № сектора.

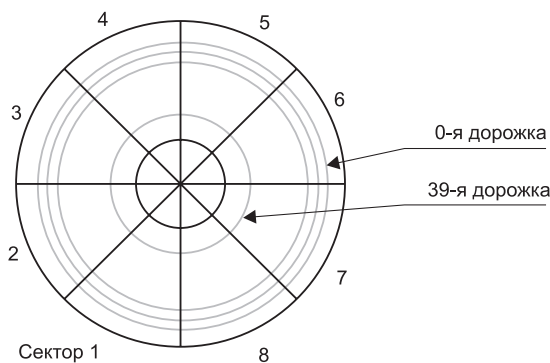


Рис. 1.5. Структура поверхности дискеты (40 дорожек, 8 секторов)

Таким образом, информационный объем дискеты равен (байт):

$$V = P \cdot D \cdot S \cdot 512,$$

где V — информационный объем дискеты, байт; P — количество поверхностей дискеты (одна или две); D — количество дорожек на поверхности; S — количество секторов на дорожке.

Если дискета является системной, то ядро MS DOS размещается начиная с 0-й дорожки, как более надежной (бóльшая длина и меньшая плотность записи).

Форматирование дискет производится при инициализации дискеты изготовителем или пользователем с помощью утилиты операционной системы.

В табл. 1.1 приведен перечень основных стандартных форматов гибких дисков, применяемых в IBM PC.

Кластер — минимальный участок на диске, который может быть выделен файловой системой для хранения одного файла. Определяется он, как правило, автоматически при форматировании винчестера по зависимости, указанной в табл. 2.3.

Таблица 1.1. Форматы гибких дисков, используемых в ПЭВМ

Количество поверхностей	Количество дорожек на поверхности	Количество секторов на дорожке	Емкость дискеты, Кбайт	Типоразмер дискеты
2	40	9	360	5.25"
2	80	9	720	3.5"
2	80	15	1200	5.25"
2	80	18	1440	3.5"
2	80	36	2880	3.5"

Размер кластера можно выбрать вручную при форматировании:

```
format d: /A:size,
```

где *size* — размер кластера в байтах.

Однако существуют некоторые правила, которых следует придерживаться: во-первых, размер кластера должен быть кратен размеру физического сектора, т. е. 512 байтам в подавляющем большинстве случаев; во-вторых, есть ограничения по количеству кластеров на разделе.

Файловые системы

Всякая операционная система создает на каждом томе (дискете, диске, пакете дисков, CD-ROM и пр.) совокупность системных данных, которая называется *файловой системой* (файловой структурой).

Файловая система (пустая) создается при инициализации (разметке) тома, затем корректируется ОС (подсистемой управления данными) при текущей работе, в процессе создания, удаления, модификации (увеличения или уменьшения объема) файлов пользователя, содержащих программы или данные.

Логическая структура файловой системы. Несмотря на большое разнообразие современных ОС и файловых систем (ФС), ими эксплуатируемых, на логическом уровне (с точки зрения приложений, которые обращаются к данным) ФС имеют сходную иерархическую структуру (рис. 1.6).

№ цилиндра (совокупность дорожек, доступных при фиксированном положении блока головок считывающего устройства), № поверхности (дорожки в цилиндре), № блока на дорожке.

Таблица содержания. Пример простейшей (абстрактной) таблицы содержания, оглавления тома (диска, пакета дисков), которая в разных ОС имеет различные наименования — VTOC — Volume Table of Content (таблица содержания тома), FAT — File Allocation Table (таблица размещения файлов), FDT — File Definition Table (таблица определения файлов) и т. п., приведен на рис. 1.7.

Таблица состоит из четырех областей:

- **область файлов.** Это таблица, имеющая обычно ограниченное (в приведенном примере $N=6$) число строк N (в MS DOS, например, $N=500$, т. е. число файлов не более 500). Количество столбцов M (в примере $M=5$) обычно выбирается из тех соображений, чтобы 85—95 % файлов, создаваемых пользователями, содержало бы не более M блоков, что зависит как от размера блока и типа пользователя, так и от общего уровня развития информационного и программного обеспечения. Первый столбец таблицы в

Имя файла (заглавная запись)		Номера блоков, выделенных для размещения файлов				
File_1		1	3	7	5	13
File_2		41	8			
File_3						
File_4		3				
Область переполнения						
File_1		23				
Список свободных блоков						
2	4	6	9	10	11	12
13						
Список сбойных блоков						
12	24	7				

Рис. 1.7. Простейшая таблица оглавления тома

каждой строке (*заглавная запись* — *Title Record*) содержит данные о файле, в данном примере — имя файла;

- *область переполнения* — дополнительная таблица аналогичной структуры, в которую записываются номера блоков особо длинных файлов (в примере — `File_1`). Организация таблицы размещения в форме области файлов и области переполнения, очевидно, позволяет сэкономить на объеме таблицы в целом, не ограничивая в то же время вероятной длины файла;
- *список свободных блоков* — необходимая информация для размещения создаваемых или расширяемых файлов. Список создается при инициализации и включает все блоки, кроме поврежденных, а затем корректируется при создании, удалении, модификации файлов;
- *список сбойных блоков* — это таблица, создаваемая при инициализации (разметке) тома (диска), пополняемая программами диагностики (примером которых может служить хорошо известный пользователям NDD — Norton Disk Doctor) и предотвращающая распределение испорченных областей на магнитном носителе под файлы данных.

Здесь не указаны такие известные атрибуты файлов, как длина (в байтах), время создания, тип (архивный, скрытый, только для чтения, не для исполнения и пр.), которые могут содержаться в заглавной записи таблицы (колонка 1 на рис. 1.7).

В развитых системах коллективного пользования такие данные содержатся в специальных таблицах разделения полномочий, поскольку перечисленные да и другие атрибуты должны быть соотнесены с конкретными пользователями.

Кроме того, где-то должны быть размещены метка тома (имя и тип/объем), количество занятого и свободного пространства и прочая совокупная информация по тому данным.

Перечислим особенности ситуации, зафиксированной на рис. 1.7 в простейшей (искусственной) файловой системе:

`File_1` занимает 6 блоков, это число больше максимального, поэтому адрес блока № 6 (23) размещен в таблице переполнения;

`File_2` занимает 2 блока, что меньше ограничения, поэтому вся информация сосредоточена в области файлов.

Имеются следующие конфликтные ситуации:

- `File_3` не содержит ни одного блока (следовательно, файл был удален, но заглавная запись сохранилась);

- File_4 и File_1 ссылаются на блок № 3. Это ошибка, поскольку каждый блок должен быть закреплен за единственным файлом;
- File_1 содержит ссылку на блок № 7, помеченный как сбойный (нечитаемый). Это приведет к невозможности корректно полностью прочитать данный файл — ситуация, знакомая каждому, работавшему с НГМД;
- в списке свободных блоков содержатся номера блоков № 12 (помеченный как сбойный) и № 13 (распределенный под File_1).

Это очевидные свидетельства начавшегося разрушения файловой системы. Перечисленные конфликты могут иметь своими источниками сбои, программные ошибки (разработчиков ОС), некорректное завершение ОС или целенаправленную деятельность вирусных или иных злонамеренных программ.

Косвенная адресация. Рассмотренный пример таблицы оглавления относится к случаю так называемой прямой адресации доступа (рис. 1.8). Здесь очевидны следующие особенности:

- таблица создается при инициализации диска и, даже будучи пустой, занимает определенный объем;
- создание файла (даже состоящего из одного байта) приводит к выделению блока и занятию строки таблицы.

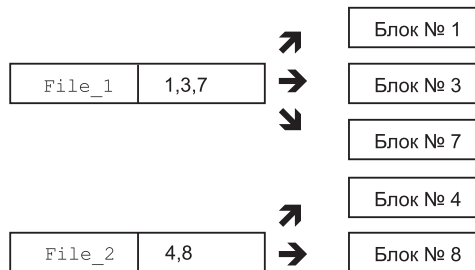


Рис. 1.8. Доступ к данным с прямой адресацией

Косвенная адресация — списковая (рис. 1.9, а) и мультисписковая (рис. 1.9, б) — создает управляющие структуры по мере необходимости (при заполнении файла). Высвобождение памяти в списковой структуре осуществляется автоматически при удалении любого промежуточного блока, содержащего также адрес последующего блока файла. Очевидно, это увеличи-

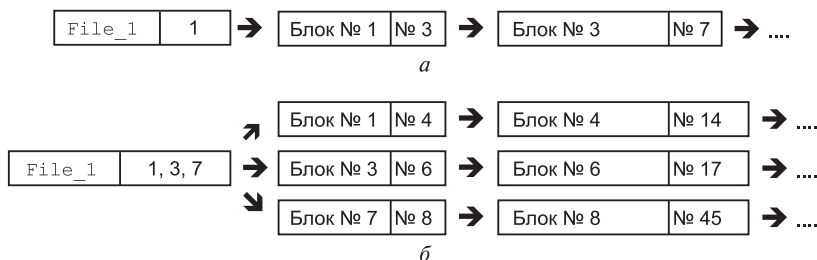


Рис. 1.9. Списковая организация доступа к данным (косвенная адресация) (а); комбинированная (мультисекторная) организация доступа (б)

вает и опасность утраты данных при ошибочном удалении или разрушении промежуточного блока файла.

Виды файлов. Все операционные системы, как правило, поддерживают следующие элементы иерархических файловых систем: обычные файлы, каталоги, специальные байт-ориентированные и блок-ориентированные файлы. Файл является массивом байтов (блоков фиксированной длины). Каталоги обеспечивают связь между именами файлов и собственно файлами. Каждый элемент каталога содержит имя файла и ссылку на конкретный файл. Для именования файлов используются корневой и текущий каталоги.

Внешние устройства (такие, как терминал, принтер) также часто представляются как файлы для упрощения работы с ними.

Устройства, на которые осуществляется вывод данных из программы или с которых осуществляется ввод (это может быть одно и то же физическое устройство, как это было в случае ранних терминалов; в современных, так называемых ANSI-терминалах, монитор и клавиатура рассматриваются как два отдельных, независимых устройства), могут быть подразделены на следующие типы:

- передачи информации битовым потоком;
- посимвольного обмена информацией;
- передачи информации порциями (записями).

Фактически это как бы «портрет» устройства, каким его «видит» прикладная программа, через посредство драйвера устройства и программ операционной системы, ответственных за ввод-вывод информации. Одно и то же устройство может быть представлено как генератор потока символов (поток-ориентированное устройство) или записей (записе-ориентированное). Поэтому, скорее, стоит говорить о типе файлов, расположенных на том или ином устройстве.

Различают следующие виды файлов:

- по типу записей:
 - файлы с записями постоянной длины;
 - файлы с записями переменной или неопределенной длины;
 - файлы, образующие байтовый или битовый поток;
- по способу выборки информации:
 - файлы последовательного доступа;
 - файлы прямого доступа;
 - файлы доступа по индексу (обычно — файлы базы данных).

Далее возникает проблема идентификации данных, размещенных на носителе (в файле). Каким образом можно правильно сопоставить тем или иным битовым комбинациям, размещенным в файле, те или иные области оперативной памяти, куда они должны считываться с носителя для последующей обработки или обновления? В частности, различные способы идентификации связаны с понятиями базового и стандартного буферизованного ввода-вывода.

Базовый ввод-вывод. Базовый уровень ввода-вывода обеспечивает обмен с файлом, интерпретируемым как одномерный массив байтов с прямым последовательным доступом.

Для каждого файла система ведет указатель чтения/записи. При чтении (записи) n байтов указатель продвигается вперед по файлу на n байтов и устанавливается в позицию очередного читаемого (записываемого) символа.

В начале работы с файлом его создают или открывают. Файлы, открытые данной программой, имеют внутреннюю (в пределах данного процесса) нумерацию начиная с 0. Системный вызов, открывающий файл, возвращает номер открытого файла, который используется при чтении и записи. После того как файл открыт, к нему могут применяться функции чтения или записи. При чтении из файла последовательно читаются очередные байты и возвращается число прочитанных байтов. Оно может оказаться меньше требуемого числа, если до конца файла осталось меньше байтов, чем требуется, или если устройство не передает такого числа байтов.

При записи в файл записываются очередные байты, расположенные в памяти процесса. Если возвращаемое после записи значение не равно числу записываемых байтов, это свидетельствует об ошибке. Если очередной записываемый байт оказывает-

ся за концом файла, то обеспечивается соответствующее увеличение размера файла.

Прямой доступ к файлу реализуется вызовом соответствующей функции, устанавливающей указатель чтения/записи в требуемую позицию. Позиционирование возможно в тех файлах, где оно допускается типом файла или природой внешнего устройства.

Процесс может управлять открытым файлом, получая и задавая значения его атрибутов, а также блокируя участки файла (или файла целиком) от доступа других процессов.

По окончании работы с файлом его следует закрыть. При завершении программы все открытые файлы закрываются автоматически.

Особенность каталога состоит в том, что запись в него может делать только система — программа может только читать элементы каталога.

Стандартный буферизованный ввод-вывод является надстройкой над базовым уровнем. Подобно базовому уровню, он интерпретирует файл (поток, в терминах данного уровня) как одномерный массив байтов с прямым доступом.

Потоки дают возможность обмениваться с файлом, буферизируя данные в памяти процесса. При чтении из потока происходит считывание блока данных из файла в буфер, а из буфера процессу передается столько байтов, сколько он запросил. Когда при очередном чтении из потока в буфере уже нет требуемых данных, происходит очередное считывание блока данных из файла в буфер. Аналогично при записи в поток передаваемые процессом данные накапливаются в буфере и передаются системе для записи в файл только после того, как буфер заполнится, при вызове специальной функции или при закрытии потока (кстати, при выключении компьютера содержимое буферов операционной системы теряется).

Когда процесс начинает работу, он получает открытыми три стандартных потока: стандартный ввод (`stdin` — в Unix, `CON` — канал 0 в MS DOS), стандартный вывод (`stdout` — в Unix, `CON` — канал 1 в MS DOS), стандартную диагностику (`stderr` — в Unix, канал 2 в MS DOS). MS DOS предоставляет дополнительно еще три стандартных потока — канал связи (`AUX` — канал 3) и стандартное устройство печати (`PRN` — канал 4). Стандартный ввод используется как устройство чтения по умолчанию, стандартный вывод — как устройство записи по умолчанию, стандартная диагностика — для вывода сообщений об ошибках.

В операционных системах средства ввода-вывода буферизированного обмена позволяют передавать символы, символьные строки, форматировать выводимую информацию. Как и на базовом уровне, возможна установка позиции в потоке.

Разделение доступа к данным в ОС

Все рассматриваемые операционные системы поддерживают операции блокировки файла для защиты доступа к нему со стороны других процессов в многозадачной среде.

Пользователи, которым разрешено входить в систему, перечислены в учетной базе пользователей. Пользователи объединены в группы; последние перечислены в учетной базе групп. Каждому пользователю и каждой группе присвоены целочисленные идентификаторы.

Входя в систему, пользователь сообщает ей свое имя, по которому определяется его идентификатор и права доступа. Вызывая команды, пользователь тем самым порождает процессы, которые наследуют его права, пользовательский и групповой идентификаторы.

С каждым файлом связана пара идентификаторов: пользовательский и групповой. Файл наследует эти идентификаторы от эффективных идентификаторов процесса, создавшего данный файл. Процесс, эффективный пользовательский идентификатор которого совпадает с пользовательским идентификатором файла, считается владельцем данного файла.

Файл можно читать, писать и выполнять. Если файл является каталогом, выполнение означает поиск в нем. Права процессов при доступе к файлу хранятся в атрибутах защиты файла. Эти атрибуты при создании файла могут быть изменены только на основе соответствующих прав.

Проверка прав происходит, когда процесс пытается открыть файл для чтения или записи, выполнить его.

Все пользователи, имеющие доступ в систему, разделены по отношению к файлу на три категории: владельцы (эффективный пользовательский идентификатор процесса совпадает с пользовательским идентификатором файла), члены группы (эффективный групповой идентификатор процесса совпадает с групповым идентификатором файла) и прочие.

Процесс может иметь зависящие от реализации привилегии, которые дают ему дополнительные права при доступе к файлу.

Если процесс не имеет привилегий, то ему разрешается доступ к файлу в трех случаях:

- процесс является владельцем файла (см. выше), и атрибуты защиты файла разрешают запрашиваемый вид действия владельцу;
- эффективный групповой идентификатор процесса совпадает с групповым идентификатором файла, и атрибуты файла разрешают запрашиваемый вид действия группе;
- атрибуты файла разрешают запрашиваемый вид действия всем процессам.

Если ни одно из условий не выполняется, то процесс не получает доступ к файлу.

Системные вызовы операционной системы обеспечивают получение информации о пользователях и группах в учетной базе (при наличии соответствующих привилегий) и получение информации о защите конкретного файла.

Форматы файлов

В зависимости от типа и назначения файлов и возможностей ОС (методов доступа) файл может передаваться в прикладную программу как целое или блоками (физическими записями) либо логическими записями (строками, словами, символами).

В системе OS/360 основную роль играли два типа файлов:

- символьные (исходные программы или данные);
- двоичные (программы в машинных кодах).

В современных системах активно используется значительно большее разнообразие файлов, из которых мы перечислим наиболее типичные файлы данных:

- *текстовые файлы* — обобщенное название для простых и размеченных текстов, ASCII-файлов и других наборов данных символьной информации, которые интерпретируются и обрабатываются текстовыми редакторами, процессорами, анализаторами (Lexicon, Word, TEC, анализаторы SGML, HTML);
- *текст без разметки* (планарный) — файл, содержащий только отображаемые (воспроизводимые на всех печатающих устройствах и терминалах) символы кода ASCII, а также простейшие управляющие символы: CR — возврат ка-

Таблица 1.2. Основные типы файлов, обрабатываемых в ПЭВМ

Тип, расширение имени	Вид информации, содержащейся в файле
exe, com	Программа, готовая к исполнению
bat	Текстовый командный файл
sys	Системный файл
ovl, ovr	Оверлейный файл
pif	Программно-информационный файл Windows
txt, lst	Текстовый файл в формате DOS
doc	Документ (чаще всего в формате WinWord)
rtf	Размеченный текстовый файл (Rich Text Format)
dot	Файл формата документа (Document Type)
pdf	Формат документа Adobe Acrobat
wri	Документ редактора Write для Windows
wps	Документ текстового процессора MS WORKS
bak, old	Старая копия файла, создаваемая перед его изменением
arj, rar, zip, lzh, ain, arc, ice, pak, zoo	Архивные файлы
bas	Текст программы на языке Basic
pas	Текст программы на языке Turbo Pascal
c	Текст программы на ЯП Си
bmp, pcx, gif, tif, jpg, ico	Графические файлы
dbf	Файлы базы данных формата DBase, FoxBase, Clipper
wdb	Базы данных формата MS WORKS
wks	Электронная таблица формата MS WORKS
xls	Электронные таблицы EXCEL
lib, dll	Файлы библиотек
dat	Файл данных
ini	Файл инициализации
hlp	Файл справки (подсказки, помощи)
ext	Файл расширений
mnu	Файл меню
wav, mid, mp3, mod	Звуковые файлы
avi, mov, mpg	Файлы видеоклипов

ретки; LF — перевод строки; TAB — символ табуляции, иногда NP — новая страница;

- *текст с разметкой* — планарный файл, содержащий бинарную и символьную разметку, управляющую отображением информации (программно и/или аппаратурно);
- *ASCII-файл* — файл, содержащий только отображаемые коды левой части кодовой таблицы ASCII (латиница и служебные символы), обычно применяется для хранения документов с символьной разметкой (RTF, SGML, HTML);
- *табличный файл* — файл, содержащий форматированные данные (символьные, численные и др), образующие строки и столбцы таблиц, создаваемых и обрабатываемых табличными СУБД (FoxPro, Clipper, MS Access) и/или табличными процессорами (SuperCalc, MS Excell и др.);
- *графический файл* — бинарный файл, содержащий графическую информацию. Форматы: TIF (Tagged Image File), BMP (Bit-Mapped Picture), а также ряд других — PCX, PICS и т. д.;
- *мультимедиа-файлы* — бинарные файлы, содержащие оцифрованную аудио- (типы WAV или MIDI-Sequencer), видео- (формат MPEG) или смешанную информацию.

В табл. 1.2 приведены основные типы файлов, используемых в ОС DOS, Windows, и соответствующие им расширения имени.

1.3. Управление заданиями (процессами, задачами)

Основными понятиями управления прохождением задач в ЭВМ являются *процесс, задача, работа, программа, ресурс, дисциплина распределения ресурса* [14].

Процесс — минимальный программный объект, обладающий собственными системными ресурсами (запущенная программа).

Классификация процессов

По временным характеристикам различают интерактивные, пакетные процессы и процессы реального времени. Время существования интерактивного процесса определяется реакцией ЭВМ на запрос обслуживания и составляет секунды. Процессы реального времени имеют гарантированное время окончания работы

и время реакции порядка миллисекунд (мс). Пакетные процессы запускаются один за другим и здесь время реакции — минуты и даже часы.

По генеалогическому признаку различают порождающие и порожденные процессы.

По результативности различают эквивалентные, тождественные и равные процессы. Все они имеют одинаковый конечный результат, но эквивалентные процессы могут реализовываться как на одном, так и на многих процессорах по одному или разным алгоритмам, т. е. они имеют разные трассы, которые определяют порядок и длительность пребывания процесса в разных состояниях. Тождественные процессы реализуются по одной и той же программе, но имеют разные трассы. Одинаковые процессы реализуются по одной программе и имеют одинаковые трассы.

По времени развития процессы делятся на последовательные, параллельные и комбинированные (для последних есть точки, в которых существуют оба процесса, и точки, в которых существует только один процесс).

По месту развития процессы делятся на внутренние (реализуются на центральном процессоре) *и внешние* (реализуются на внешних процессорах).

По принадлежности к операционной системе процессы бывают *системные* (исполняют программу из состава операционной системы) и *пользовательские*.

По связности различают процессы:

- а) взаимосвязанные, которые имеют какую-то связь (пространственно-временную, управляющую, информационную);
- б) изолированные — слабо связанные;
- в) информационно-независимые, которые используют совместные ресурсы, но имеют собственные информационные базы;
- г) взаимодействующие — имеют информационные связи и разделяют общие структуры данных;
- д) взаимосвязанные по ресурсам;
- е) конкурирующие.

Порядок взаимосвязи процессов определяется правилами синхронизации. Процессы могут находиться в отношении:

- а) предшествования — один всегда находится в активном состоянии раньше, чем другой;
- б) приоритетности — когда процесс может быть переведен в активное состояние только в том случае, если в состоянии готов-

ности нет процессов с более высоким приоритетом, или процессор свободен, или на нем реализуется процесс с меньшим приоритетом;

в) взаимного исключения — в процессе используется общий критический ресурс, и процессы не могут развиваться одновременно: если один из них использует критический ресурс, то другой находится в состоянии ожидания.

Ресурс — любая потребляемая (расходуемая) сущность. По запасам ресурсы подразделяются на исчерпаемые и неисчерпаемые. Потребители ресурсов — процессы. Ресурс — средство вычислительной системы, которое может быть выделено процессу на определенный интервал времени.

Процессор — любое устройство в составе ЭВМ, способное автоматически выполнять допустимые для него действия (процессы, каналы и устройства, работающие с каналами). Реализация системы управления процессами в составе ОС предъявляет определенные требования к свойствам процессоров.

Классификация ресурсов

По признаку реальности ресурсы делятся на *физические* и *виртуальные* (последние только в отдельных свойствах схожи с физическими ресурсами).

По возможности расширения свойств делятся на *эластичные* и *жесткие* (не допускающие виртуализации).

По степени активности разделяются на *пассивные* и *активные* (могут выполнять действия по отношению к другим ресурсам).

По времени существования — *постоянные* (доступны во все время процесса: и до, и после его работы) и *временные*.

По степени важности — *основные* и *второстепенные* (допускают альтернативное развитие процесса при их отсутствии).

По функциональной избыточности при распределении — *дорогой*, но предоставляемый быстро, и *дешевый*, но предоставляемый с ожиданием.

По структуре — *простые* (не содержат составных элементов) и *составные*. Они различаются числом состояний: простой может быть только в двух состояниях — доступен или занят.

По характеру использования распределяемых ресурсов — *потребляемые* и *воспроизводимые* ресурсы (допускают многократное использование и освобождение).

По характеру использования — последовательно и параллельно используемые (используются несколькими процессами).

По форме реализации — жесткие (в принципе не допускают копирования) и *мягкие* (допускают тиражирование и подразделяются на программные и информационные ресурсы).

Дисциплина распределения ресурса определяет порядок использования многими процессами того или иного ресурса, который в каждый момент времени может обслуживать только один процесс.

Управление процессами

Процесс — это программный модуль, выполняемый в центральном процессоре (CPU). Операционная система контролирует следующую деятельность, связанную с процессами:

- создание и удаление процессов;
- планирование процессов;
- синхронизацию процессов;
- коммуникацию процессов;
- разрешение тупиковых ситуаций.

Не следует смешивать понятия «процесс» и «программа». Программа — это план действий, а процесс — это **с о б с т в е н н о д е й с т в и е**, поэтому понятие процесса включает:

- программный код;
- данные;
- содержимое стека;
- содержимое адресного и других регистров процессора.

Таким образом, для одной программы могут быть созданы несколько процессов в том случае, если с помощью одной программы в CPU выполняется несколько несовпадающих последовательностей команд. За время существования процесс многократно изменяет свое состояние.

Различают следующие состояния процесса (рис. 1.10):

- *новый* (процесс только что создан);
- *выполняемый* (команды программы выполняются в CPU);
- *ожидающий* (процесс ожидает завершения некоторого события, чаще всего операции ввода-вывода);
- *готовый* (процесс ожидает освобождения CPU);
- *завершенный* (процесс завершил свою работу).

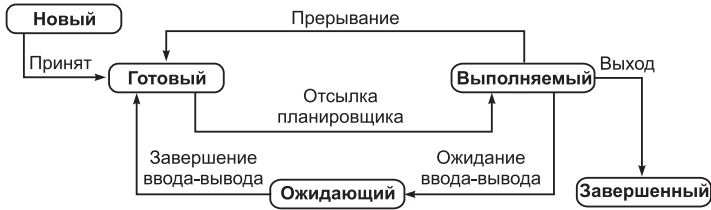


Рис. 1.10. Состояния процесса

Переход из одного состояния в другое не может выполняться произвольным образом. На рис. 1.10 приведена типовая диаграмма переходов для состояний процессов.

Каждый процесс представлен в операционной системе набором данных, называемых *таблицей управления процессом* (ТУП — РСВ — process control block). В РСВ процесс описывается набором значений, параметров, характеризующих его текущее состояние и используемых операционной системой для управления прохождением процесса через компьютер.

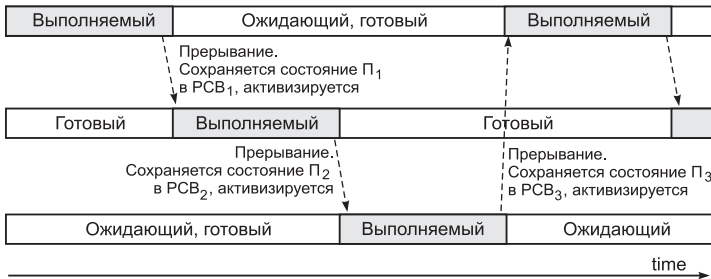


Рис. 1.11. Переходы между процессами

На рис. 1.11 схематически показано, каким образом операционная система использует РСВ для переключения процессора с одного процесса на другой.

Планирование процессов

Система управления процессами обеспечивает прохождение процесса через компьютер. В зависимости от состояния процесса ему должен быть предоставлен тот или иной ресурс. Например, новый процесс необходимо разместить в основной памяти, следовательно, ему необходимо выделить часть адресного про-

странства. Процессу в состоянии готовый должно быть предоставлено процессорное время. Выполняемый процесс может потребовать оборудование ввода-вывода и доступ к файлу [14].

Очереди. Распределение процессов между имеющимися ресурсами носит название *планирование процессов*. Одним из методов планирования процессов, ориентированных на эффективную загрузку ресурсов, является метод очередей ресурсов. Новые процессы находятся во входной очереди, часто называемой *очередью работ — заданий*.

Входная очередь располагается во внешней памяти, во входной очереди процессы ожидают освобождения ресурса — адресного пространства основной памяти.

Готовые к выполнению процессы располагаются в основной памяти и связаны *очередью готовых процессов*. Процессы в этой очереди ожидают освобождения ресурса *процессорное время*.

Процесс в состоянии ожидания завершения операции ввода-вывода находится в одной из *очередей к оборудованию ввода-вывода*.

При прохождении через компьютер процесс мигрирует между различными очередями под управлением программы, которая называется *планировщик* (scheduler).

Операционная система, обеспечивающая режим мультипрограммирования, обычно включает два планировщика — *долгосрочный* и *краткосрочный*. Например, в OS/360 долговременный планировщик назывался *планировщиком заданий*, а краткосрочный — *супервизором задач*.

На уровень долгосрочного планирования выносятся редкие системные действия, требующие больших затрат системных ресурсов, на уровень краткосрочного планирования — частые и более короткие процессы. На каждом уровне существует свой объект и собственные средства управления им.

Основное различие между долгосрочным и краткосрочным планировщиками заключается в частоте запуска, например, краткосрочный планировщик может запускаться каждые 100 мс, долгосрочный — 1 раз за несколько минут.

Долгосрочный планировщик решает, какой из процессов, находящихся во входной очереди, должен быть переведен в очередь готовых процессов в случае освобождения ресурсов памяти.

Долгосрочный планировщик выбирает процесс из входной очереди с целью создания неоднородной мультипрограммной смеси. Это означает, что в очереди готовых процессов должны

находиться в разной пропорции как процессы, ориентированные на ввод-вывод, так и процессы, ориентированные на преимущественную работу с CPU.

На уровне долгосрочного планирования объектом является не отдельный процесс, а некоторое объединение процессов по функциональному назначению, которое называется работой (приложением). Каждая работа рассматривается как независимая от других работ деятельность, связанная с использованием одной или многих программ и характеризующаяся конечностью и определенностью. По мере порождения новых работ создается собственная виртуальная машина для их выполнения. Например, в ОС Windows 95 для каждого 32-разрядного приложения реализуется своя виртуальная машина. Распределение машин производится однократно в отличие от краткосрочного планирования, где процессор процессу может выделяться многократно.

Краткосрочный планировщик решает, какой из процессов, находящихся в очереди готовых процессов, должен быть передан на выполнение в CPU. В некоторых операционных системах долгосрочный планировщик может отсутствовать. Например, в *системах разделения времени* (time-sharing system) каждый новый процесс сразу же помещается в основную память.

На уровне краткосрочного планирования объектом управления являются процессы, которые выступают как потребители центрального процессора для внутренних процессов или внешнего процессора для внешних процессов. Причинами порождения процесса могут быть процессы на том же уровне или сигналы, посылаемые от долгосрочного планировщика.

Выделение процессора процессу производится многократно, с целью достижения эффекта мультипрограммирования, и такой процесс называется диспетчеризацией.

Взаимодействие процессов

Совместно выполняемые процессы могут быть либо *независимыми*, либо *взаимодействующими*. Взаимодействие процессов часто понимается в смысле взаимного обмена данными через общий буфер данных.

Взаимодействие процессов удобно рассматривать в схеме производитель-потребитель. Например, программа вывода на печать производит последовательность символов, которые потреб-

ляются драйвером принтера, или компилятор производит ассемблерный текст, который затем потребляется ассемблером.

Для взаимодействия процесса-производителя и процесса-потребителя создается совместный буфер, заполняемый процессом-производителем и потребляемый процессом-потребителем.

Буфер имеет фиксированные размеры и, следовательно, процессы могут находиться в состоянии ожидания, когда:

- буфер заполнен — ожидает процесс-производитель;
- буфер пуст — ожидает процесс-потребитель.

Буфер может предоставляться и поддерживаться самой ОС, например с помощью средств межпроцессной коммуникации, либо должен быть организован прикладным программистом. При этом оба процесса используют общий участок памяти.

Взаимодействие заключается в передаче данных между процессами или совместном использовании некоторых ресурсов и обычно реализуется с помощью таких механизмов, как **транспортеры, очереди, сигналы, семафоры**.

Транспортеры (каналы). Являются средством взаимодействия родственных процессов, представляют собой область памяти, имеющую файловую организацию, для которой обеспечивается запись и считывание данных в транспортере. Реализуется очередь обслуживания. Порядок записи данных на транспортер неизменен, не допускается повторное считывание данных. Обмен данными происходит не непосредственно, а через транспортер. Из вызвавшего процесса задается размер транспортера. Дочерние процессы могут использовать родительский транспортер.

Очереди. Эти механизмы могут обеспечивать передачу или использование общих данных без перемещения данных, а с передачей элемента очереди, содержащего указатель данных и объем массива данных. Очередь используется вместе с механизмом общей памяти. Элемент очереди может быть считан с уничтожением или без уничтожения этого элемента. Чтение элемента может осуществляться в соответствии с механизмом очереди или стека. Чтение элементов очереди осуществляет только создающий очереди процесс, все другие процессы могут только записать элемент в очередь. Создающий процесс может выполнять следующие действия над очередью:

- создание очереди;
- просмотр очереди;
- чтение очереди;
- закрытие очереди.

Записывающий процесс осуществляет действия:

- открытие очереди;
- запись в очередь;
- закрытие очереди.

Имя очереди, которое присваивается создающим процессом, имеет вид полной спецификации файла. Ожидание элементов в очереди организуется с помощью семафора, сигнализирующего о записи элемента в очередь. Для работы с очередью определены такие дополнительные функции:

- определение количества элементов в очереди в текущий момент;
- очистка очереди создавшим ее процессом.

Преимущества очереди: передача данных по указателю без копирования, гибкое изменение порядка передачи и доступа, возможность просмотра элементов очереди без их удаления.

Сигналы. Сигнал является механизмом передачи требования от одного процесса к другому на немедленное выполнение действия. Обработчик сигнала создается процессом и помещается в начале первого потока процесса. Является аналогом обработки прерывания. При передаче управления обработчику передается адрес возврата и тип принятого сигнала. Процесс, посылающий сигнал типа флаг, может передать дополнительную информацию обработчику сигнала. Характер выполняемых действий при возникновении сигнала: обработка системной ошибки при появлении сигнала, блокирование сигнала, передача управления подпрограмме.

Семафоры. Являются механизмами передачи сообщений от одного потока к другому о наступлении некоторого события. Различают семафоры системные и оперативной памяти. Семафоры оперативной памяти — двойное слово в памяти системы, его описатель — адрес места в памяти. Такие семафоры не создаются и не открываются, а устанавливаются в определенное состояние. Процессы, использующие семафоры оперативной памяти, должны иметь доступ к соответствующему сегменту памяти. Операционная система такие семафоры не обслуживает и не сообщает об их освобождении или захвате. При создании семафора или его открытии возвращается описатель семафора, включающий его имя. Операционная система контролирует завершение каждого процесса, владеющего системным семафором, и освобождает его для процессов.

Если семафор свободен, то он захватывается вызывающим его процессом, если семафор занят, то вызвавший его поток пе-

переходит в режим ожидания освобождения семафора или ожидает истечения времени. Если семафор освобождается всеми использующими его процессами, то он удаляется из системы.

Управление семафором реализуется с помощью функций:

- установки семафора с целью сигнализации;
- ожидания вызывающим потоком, пока семафор не будет выключен;
- ожидания потоком выключения одного из нескольких семафоров.

Операционные системы используют разные термины для определения способов межпроцессного взаимодействия.

В операционных системах OS/2 и Microsoft Windows существует специальный механизм для взаимодействия процессов в реальном масштабе времени. Этот механизм называется DDE (*Dynamic Data Exchange* — динамический обмен данными). Он стандартизирует процесс обмена командами, сообщениями и объектами для обработки между задачами. Наиболее распространенным процессом, для которого используется DDE, является печать.

Другим интерфейсом для обмена данными является OLE (*Object Linking and Embedding* — связывание и встраивание объектов). Этот интерфейс позволяет хранить объекты, созданные одной программой, в объектах, созданных другой программой, а также редактировать (печатать) их без нарушения целостности информации и связей.

Одним из наиболее простых, удобных и интуитивных интерфейсов межпрограммного взаимодействия является буфер обмена — Clipboard. Буфер обмена может содержать в себе один информационный объект — фрагмент текста, рисунок и т. д. С помощью системного вызова процесс может получить копию информации, содержащейся в буфере обмена, или сам поместить объект в буфер, при этом старое содержимое буфера теряется. Таким образом, программы получают простой, но эффективный способ обмена информацией в процессе своей работы.

Планирование работы процессора

Краткосрочный планировщик выбирает процессы из очереди готовых процессов и передает их на выполнение в CPU. Существуют различные алгоритмы или стратегии решения этой задачи, различающиеся отношением к критериям планирования.

Известны следующие критерии, позволяющие сравнивать алгоритмы краткосрочных планировщиков:

- утилизация CPU (использование процессора). Утилизация CPU теоретически может находиться в пределах от 0 до 100 %. В реальных системах утилизация CPU колеблется в пределах 40 % для легко загруженного CPU, 90 % для тяжело загруженного CPU;
- пропускная способность CPU. Пропускная способность CPU может измеряться количеством процессов, которые выполняются в единицу времени;
- время оборота. Для некоторых процессов важным критерием является полное время выполнения, т. е. интервал от момента появления процесса во входной очереди до момента его завершения. Это время названо временем оборота и включает время ожидания во входной очереди, время ожидания в очереди готовых процессов, время ожидания в очередях к оборудованию, время выполнения в процессоре и время ввода-вывода;
- время ожидания — под этим понимается суммарное время нахождения процесса в очереди готовых процессов;
- время отклика — для интерактивных программ важным показателем является время отклика или время, прошедшее от момента попадания процесса во входную очередь до момента первого обращения к терминалу.

Очевидно, что простейшая стратегия краткосрочного планировщика должна быть направлена на максимизацию средних значений загруженности и пропускной способности, времени ожидания и времени отклика.

В ряде случаев используются сложные критерии, например так называемый минимаксный критерий, т. е. вместо простого критерия минимум среднего времени отклика используется *минимум максимального* времени отклика.

Стратегии планирования процессора

«*Первый пришел — первый обслуживается*» (FIFO — first in, first out, или FCFS — first come, first served). FCFS является наиболее простой стратегией планирования процессов и заключается в том, что процессор передается тому процессу, который раньше всех других его запросил.

Когда процесс попадает в очередь готовых процессов, УТП (PCB) присоединяется к хвосту очереди.

Среднее время ожидания для стратегии FCFS часто весьма велико и зависит от порядка поступления процессов в очередь готовых процессов.

Стратегии FCFS присущ так называемый «эффект конвоя». В том случае, когда в компьютере имеется один большой процесс и несколько малых, то все процессы собираются в начале очереди готовых процессов, а затем в очереди к оборудованию. Таким образом, «эффект конвоя» приводит к снижению загрузки как процессора, так и периферийного оборудования.

Стратегия «наиболее короткая работа выполняется первой», SJF — Shortest Job First. Одним из методов борьбы с «эффектом конвоя» является стратегия, позволяющая процессу из очереди выполняться первым. Стратегия SJF снижает время ожидания очереди. Наибольшая трудность в практической реализации SJF заключается в невозможности заранее определить величину времени последующего обслуживания.

Поэтому стратегия SJF часто применяется в долгосрочных планировщиках, обслуживающих пакетный режим. В этом случае вместо величины времени последующего обслуживания используется допустимое максимальное время выполнения задания, которое программист должен специфицировать перед отправкой задания в пакет.

Приоритетное планирование. Описанные ранее стратегии могут рассматриваться как частные случаи стратегии приоритетного планирования. Эта стратегия предполагает, что каждому процессу приписывается приоритет, определяющий очередность предоставления ему CPU. Например, стратегия FCFS предполагает, что все процессы имеют одинаковые приоритеты, а стратегия SJF предполагает, что приоритет есть величина, обратная времени последующего обслуживания.

Обычно приоритет — это целое положительное число, находящееся в некотором диапазоне, например от 0 до 7 или от 0 до 1024. Будем считать, что чем меньше значение числа, тем выше приоритет процесса. Приоритеты назначаются, исходя из совокупности внутренних и внешних по отношению к операционной системе факторов.

Внутренние факторы:

- требования к памяти;
- количество открытых файлов;

- отношение среднего времени ввода-вывода к среднему времени использования ресурсов CPU и т. д.

Внешние факторы:

- важность процесса;
- тип и величина файлов, используемых для оплаты;
- отделение, выполняющее работы, и т. д.

Внутренние факторы могут использоваться для автоматического назначения приоритетов самой операционной системой, а внешние — для принудительного, с помощью оператора.

Главный недостаток приоритетного планирования заключается в возможности блокирования на неопределенно долгое время низкоприоритетных процессов.

Известен случай, когда в 1973 г. в Массачусетском технологическом институте при остановке компьютера IBM 7094 в очереди готовых процессов были обнаружены процессы, активизированные в 1967 г., но так и не выполненные.

Для устранения отмеченного недостатка используются следующие методы: процессы, время ожидания которых превышает фиксированную величину, например 15 минут, автоматически получают единичное приращение приоритета.

«Карусельная» стратегия планирования RR-Round Robin применяется в системах разделения времени. Определяется небольшой отрезок времени t_k , названный квантом времени (10... 100 мс). Очередь готовых процессов рассматривается как кольцевая. Процессы циклически перемещаются по очереди, получая CPU на время, равное одному кванту. Новый процесс добавляется в хвост очереди. Если процесс не завершился в пределах выделенного ему кванта времени, его работа принудительно прерывается, и он перемещается в хвост очереди.

Свойства стратегии Round Robin сильно зависят от величины временного кванта t_k . Чем больше временной квант, тем ближе стратегия Round Robin приближается к FCFS-стратегии. При очень малых значениях временного кванта Round Robin-стратегию называют разделением процессора — processor sharing. Теоретически это означает, что каждый из N процессов работает со своим собственным процессором, производительность процессора равна $1/N$ от производительности физического процессора.

Планирование с использованием многоуровневой очереди (Multi-level queue scheduling). Эта стратегия разработана для ситуации, когда процессы могут быть легко классифицированы на несколько групп, например часто процессы разделяют на две груп-

пы: интерактивные (процессы переднего плана) и пакетные (фоновые).

Интерактивные и пакетные процессы имеют различные требования к краткосрочному планировщику, например по отношению ко времени отклика.

Стратегия многоуровневой очереди разделяет очередь готовых процессов на несколько очередей, в каждой из которых находятся процессы с одинаковыми свойствами, и каждый из которых может планироваться индивидуальной стратегией, например Round Robin стратегия для интерактивных процессов и FCFS для пакетных процессов.

Взаимодействие очередей осуществляется по следующим правилам: ни один процесс с более низким приоритетом не может быть запущен, пока не выполняются процессы во всех очередях с более высоким приоритетом.

Работа процесса из очереди с более низким приоритетом может быть приостановлена, если в одной из очередей с более высоким приоритетом появился процесс.

Использование многоуровневой очереди с обратными связями (multilevel feedback queue scheduling) (рис. 1.12). Обычная многоуровневая очередь не допускает перемещения процессов между очередями. Многоуровневая очередь с обратными связями предполагает, что процессы при определенных условиях могут перемещаться между очередями. Здесь организуется N очередей. Все новые запросы поступают в конец первой очереди. Первый запрос из i -й очереди поступает на обслуживание лишь тогда, когда все очереди от 1-й до $i-1$ -й пустые. На обслуживание выделяется квант времени t_k . Если за это время обслуживание запро-

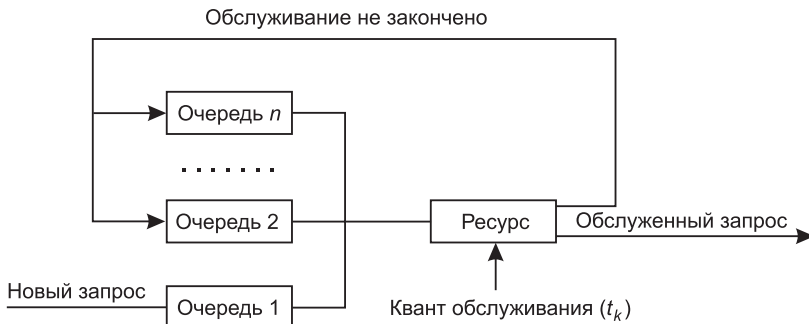


Рис. 1.12. Многоуровневая очередь с обратными связями

са завершается полностью, то он покидает систему. В противном случае недообслуженный запрос поступает в конец $i + 1$ -й очереди.

После обслуживания запроса из i -й очереди система выбирает для обслуживания запрос из непустой очереди с самым младшим номером. Таким запросом может быть следующий запрос из очереди i или из очереди $i + 1$ (при условии, что после обслуживания запроса из очереди i последняя оказалась пустой). Новый запрос поступает в 1-ю очередь ($i = 1$). В такой ситуации после окончания времени t_k , выделенного для обслуживания запроса из очереди i , будет начато обслуживание запроса первой очереди. Если система выходит на обслуживание заявок из N -й очереди, то они обслуживаются либо по дисциплине FIFO (каждая заявка обслуживается до конца), либо по циклическому алгоритму. Данная система наиболее быстро обслуживает все короткие по времени обслуживания запросы. Недостаток системы заключается в затратах времени на перемещение запросов из одной очереди в другую.

Данная стратегия является универсальной и сочетает в себе свойства всех рассмотренных раньше стратегий — FCFS, SJF, приоритетная, Round Robin, многоуровневая очередь.

Приоритетная многоочередная дисциплина обслуживания (рис. 1.13). Вновь поступающие в систему запросы устанавливаются не обязательно в 1-ю очередь, а в очередь в соответствии с имеющимися приоритетами, которые определяются параметрами обслуживания процессов. Приоритетные многоочередные

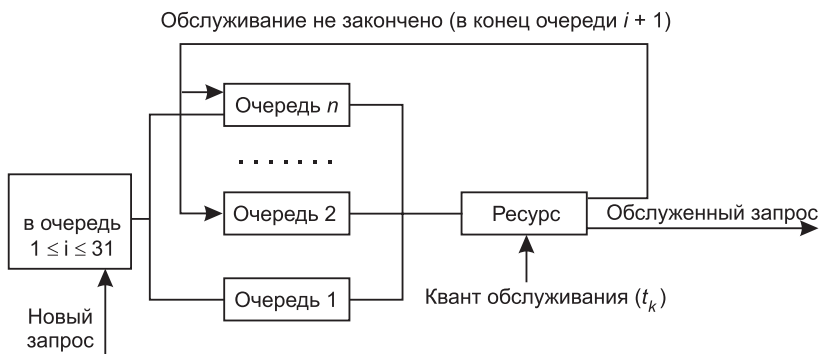


Рис. 1.13. Многоочередная с приоритетами система обслуживания

дисциплины обслуживания могут использовать обслуживание с абсолютным и относительным приоритетом. При обслуживании с абсолютным приоритетом приоритет определяется номером очереди, и первыми обслуживаются запросы, обладающие наивысшим приоритетом (из очереди с меньшим номером запрос из очереди $i - 1$ будет прерывать обработку запроса из очереди i).

В данной дисциплине еще более увеличивается степень дискриминации по среднему времени ожидания в очереди между высоко- и низкоприоритетными запросами. Время ожидания высокоприоритетных заявок сокращается, но за счет большей задержки в обслуживании низкоприоритетных заявок. Достигается это за счет усложнения логики системы, дополнительной обработки запросов и выбора правила дообслуживания прерываемых процессов. Обслуживание с относительным приоритетом не вызывает прерывания обслуживаемой заявки до ее завершения, даже если она менее приоритетная.

Управление неvirtуальной памятью

Свопинг (swapping). Свопингом (перекачкой) называется метод управления памятью, основанный на том, что все процессы, участвующие в мультипрограммной обработке, хранятся во внешней памяти.

Процесс, которому выделен CPU, временно *перемещается в основную память* (swar in/roll in). В случае прерывания работы процесса он *перемещается обратно во внешнюю память* (swar out/roll out)*.

Свопинг иногда используют при приоритетном планировании CPU. В этом случае с целью освобождения памяти для высокоприоритетных процессов, низкоприоритетные процессы перемещаются во внешнюю память.

Основное применение свопинг находит в системах разделения времени, где он используется одновременно со стратегией Round Robin планирования CPU.

В начале каждого временного кванта блок управления памятью выгружает из основной памяти процесс, работа которого была только что прервана, и загружает очередной выполненный

* При свопинге из основной памяти во внешнюю (и обратно) перемещается вся программа, а не ее отдельная часть.

процесс. Метод свопинга влияет на величину временного кванта стратегии Round Robin.

Пример. Пусть очередной загружаемый в память процесс имеет размер 100 Кбайт. Диск позволяет читать данные со скоростью 1 Мбайт в секунду, следовательно, 100 Кбайт могут быть загружены за 100 мс. Будем считать, что для первоначального подвода головки чтения-записи потребуется 8 мс. Таким образом, операция свопинг займет 108 мс, а общее время свопинга — 216 мс.

Для эффективной загруженности процессора время свопинга должно быть существенно меньше времени счета. Следовательно, для рассмотренного примера квант времени должен быть существенно больше, чем 216 мс. Ясно, что это число значительно увеличится, если перемещаемый процесс имеет размер, например, 1 Мбайт.

Недостаток «чистого» свопинга заключается в больших потерях времени на загрузку или выгрузку процессов. Поэтому в современных операционных системах используются модифицированные варианты свопинга.

Так, например, во многих версиях операционной системы Unix свопинг включается только в том случае, когда количество процессов в памяти становится слишком большим.

Смежное размещение процессов. Методы размещения процессов в основной памяти по отношению к расположению участков памяти, выделенных для одной и той же программы, делят на два класса. Первый — *метод смежного размещения*, второй — *метод несмежного размещения*.

Смежное размещение является простейшим и предполагает, что в памяти, начиная с некоторого начального адреса, выделяется один непрерывный участок адресного пространства.

При несмежном размещении программа разбивается на множество частей, которые располагаются в различных, необязательно смежных участках адресного пространства.

Однопрограммный режим. Рисунок 1.14 иллюстрирует смежное размещение одной программы в основной памяти.

При смежном размещении размер загружаемой программы ограничивается размером ОЗУ. Для того чтобы при смежном размещении загружать программы, размеры которых превышают размеры ОЗУ, используют метод оверлейных сегментов (overlay segments) — рис. 1.15.



Рис. 1.14. Однопрограммный режим

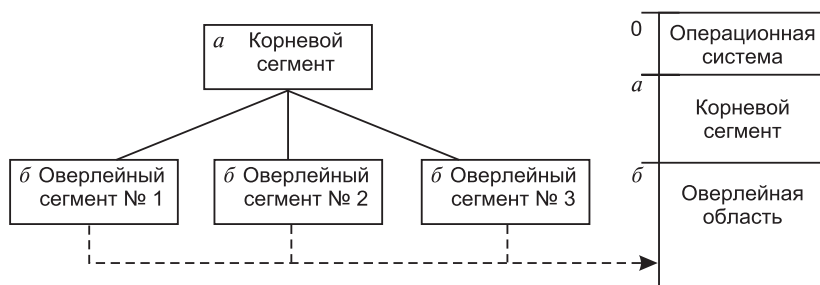


Рис. 1.15. Оверлейные структуры программ

В программе, имеющей древовидную структуру, модули второго уровня работают строго последовательно, поэтому в памяти может находиться только один из них.

Оверлейную структуру программы и последовательность загрузки оверлейных сегментов планирует сам программист.

В процессе выполнения программы все ее адреса не должны быть меньше числа a . В противном случае возможна запись какого-либо результата работы программы (поверх операционной системы) и уничтожение некоторых ее частей. Защиту операционной системы в случае смежного размещения при однопрограммном режиме можно осуществить с помощью регистра границы (рис. 1.16).

Во время работы прикладной программы все адреса, генерируемые CPU, сравниваются с содержимым регистра границы. Если генерируется адрес меньше числа a , работа программы прерывается.

Мультипрограммирование с фиксированными разделами (MFT — Multiprogramming with a fixed number of

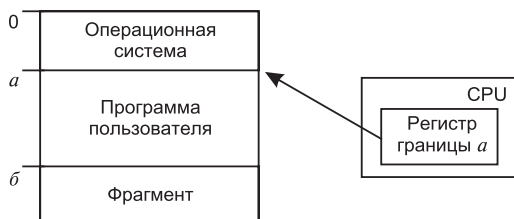


Рис. 1.16. Регистр границы

tasks) предполагает деление адресного пространства на ряд разделов фиксированного размера. В каждом разделе размещается один процесс (рис. 1.17).

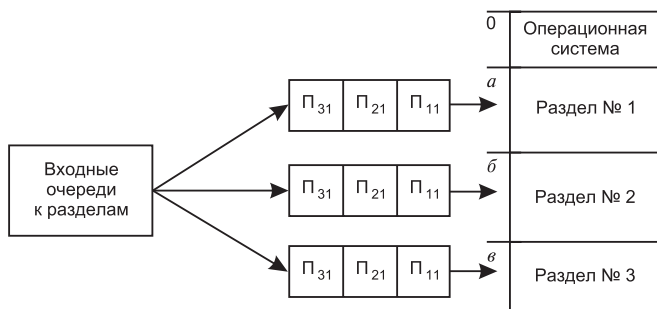


Рис. 1.17. Режим мультипрограммирования с фиксированным количеством разделов (задач)

В этом случае, если соответствующий адресам процесса раздел занят, процесс остается в очереди во внешней памяти даже в том случае, когда другие разделы свободны.

Уменьшить фрагментацию памяти (рис. 1.18) при мультипрограммировании с фиксированными разделами можно, если загрузочные модули создаются в перемещаемых адресах. Такой модуль может быть загружен в любой свободный раздел после соответствующей настройки.

При мультипрограммировании с трансляцией в перемещаемых адресах имеются две причины фрагментации. Первая — размер загруженного процесса меньше размера, занимаемого разделом (внутренняя фрагментация), вторая — размер процесса в очереди больше размера свободного раздела, и этот раздел остается свободным (внешняя фрагментация).

Для защиты памяти при мультипрограммировании с фиксированным количеством разделов необходимы два регистра. Пер-

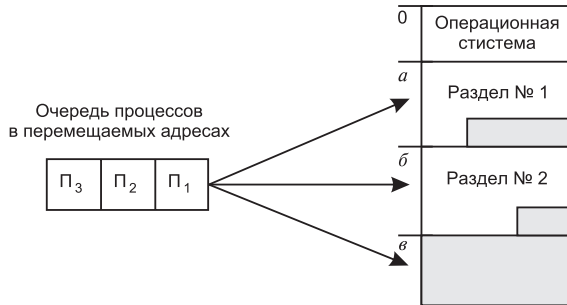


Рис. 1.18. Режим мультипрограммирования с фиксированным количеством разделов (фрагментация памяти)

вый — регистр верхней границы (наименьший адрес), второй — регистр нижней границы (наибольший адрес) — рис. 1.19.

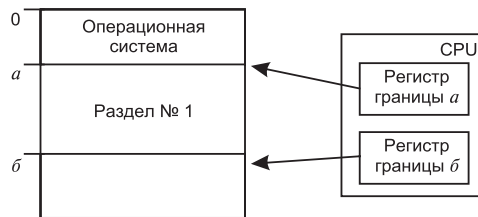


Рис. 1.19. Регистры границ для MFT

Прежде чем программа в разделе N начнет выполняться, ее граничные адреса загружаются в соответствующие регистры. В процессе работы программы все формируемые ею адреса контролируются на удовлетворение неравенства $a < \text{Адр.} < b$.

При выходе любого адреса программы за отведенные ей границы работа программы прерывается.

Мультипрограммирование с переменными разделами (multiprogramming with a variable number of tasks — MVT) предполагает деление памяти на разделы и использование загрузочных модулей в перемещаемых адресах, однако границы разделов не фиксируются.

В начальной фазе отсутствует фрагментация, связанная с тем, что размер очередного процесса меньше размера, занимаемого этим процессом раздела. На этой фазе причиной фрагментации является несоответствие размера очередного процесса и оставшегося участка памяти. По мере завершения работы про-

граммы освобождаются отдельные разделы. В том случае, когда освобождаются смежные разделы, границы между ними удаляются и разделы объединяются.

За счет объединения или слияния смежных разделов образуются большие фрагменты, в которых можно разместить большие программы из очереди. Таким образом, на фазе повторного размещения действуют те же причины фрагментации, что и для метода MFT.

Мультипрограммирование с переменными разделами и уплотнением памяти. Ясно, что этот метод может создать ситуацию, когда в памяти образуется множество малых фрагментов, каждый из которых может быть недостаточен для размещения очередного процесса, однако суммарный размер фрагментов превышает размер этого процесса.

Уплотнением памяти называется перемещение всех занятых разделов по адресному пространству памяти таким образом, чтобы свободный фрагмент занимал одну связную область (рис. 1.20).



Рис. 1.20. Режим уплотнения памяти

На практике реализация уплотнения памяти сопряжена с усложнением операционной системы и обладает следующими недостатками:

- в тех случаях, когда мультипрограммная смесь неоднородна по отношению к размерам программ, возникает необходимость в частом уплотнении, что *расходует ресурс процессорного времени и компенсирует экономию ресурса памяти*;
- во время уплотнения все прикладные программы переводятся в состояние «ожидание», что приводит к невозможности выполнения программ в реальном масштабе времени.

Основные стратегии заполнения свободного раздела. Рассмотренные методы мультипрограммирования предполагают наличие входной очереди/очередей к разделам основной памяти.

В том случае, когда освобождается очередной раздел, операционная система должна выбрать один из процессов для размещения его в памяти. Алгоритм выбора может использовать одну из следующих трех стратегий:

- стратегия наиболее подходящего — выбирает процесс, которому в освободившемся разделе наиболее тесно (выигрыш в памяти);
- стратегия первого подходящего — выбирает первый процесс, который может разместить в освободившемся разделе;
- стратегия наименее подходящего — выбирает процесс, которому в освободившемся разделе наиболее свободно (в этом случае остающийся фрагмент часто достаточен для размещения еще одного процесса).

Страничная организация памяти

Страничная организация памяти (paging) относится к методам несмежного размещения процессов в основной памяти.

Основное достоинство страничной организации памяти заключается в том, что она позволяет свести к минимуму общую фрагментацию за счет полного устранения внешней фрагментации и минимизации внутренней фрагментации.

Базовый метод. Адресное пространство основной и внешней памяти разбивают на блоки фиксированного размера, называемые *страничными рамками* (frames). Логическое адресное пространство программы также разбивается на блоки фиксированного размера, называемые *страницами* (pages). Размеры страничных рамок и страниц совпадают. Процесс загружается в память постранично, причем каждая страница помещается в любую свободную страничную рамку основной памяти.

Каждый адрес, генерируемый процессором, состоит из двух частей: N — номер страницы (page number) и C — смещение в пределах страницы (offset). Номер страницы может использоваться как индекс для таблицы страниц (page table).

Таблица страниц содержит начальные адреса f всех страничных рамок, в которых размещена программа. Физический адрес

определяется путем сложения начального адреса страничной рамки f и смещения c (рис. 1.21, 1.22).

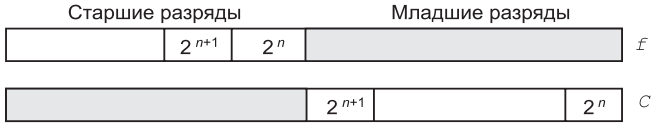


Рис. 1.21. Структура адреса при страничной организации

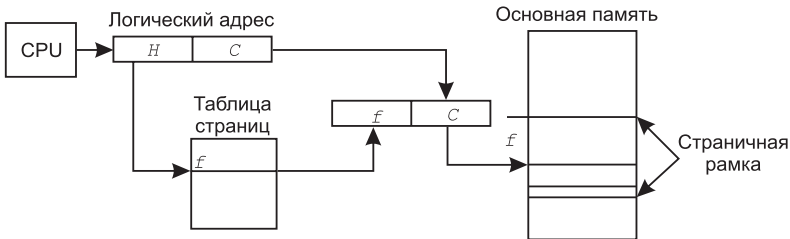


Рис. 1.22. Генерация физического адреса

Рисунок 1.23 показывает, что страничная организация памяти полностью исключает внешнюю фрагментацию. Внутренняя фрагментация не превышает величины $\text{page\_size} - \text{QElem}$, где page_size — размер страничной рамки, а QElem — минимальный адресуемый элемент основной памяти.

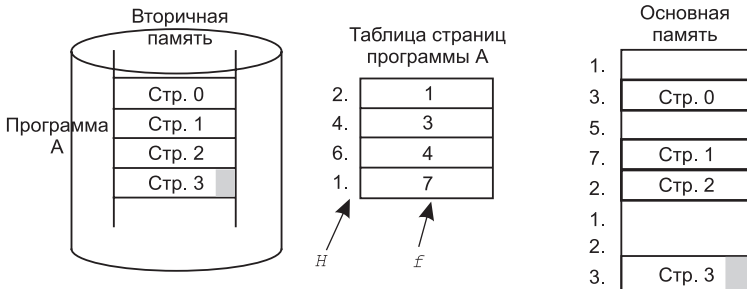


Рис. 1.23. Размещение содержания страниц на внешней памяти

Для ускорения вычисления физического адреса операцию суммирования заменяют операцией конкатенации.

На рисунке заштрихованы незаполненные нулевыми разрядами. Для того чтобы операция конкатенации была возможна, необходимо, чтобы базовые адреса страничных рамок располагались

только в старших разрядах (2^{n+1}), а следующие — только в младших разрядах (2^0 , 2^1 , 2^2).

Например, при $n = 9$ базовые адреса страничных рамок — это следующий ряд: 512, 1024, 1536. Следовательно, размер страничной рамки равен 512 байт. В современных операционных системах типичный размер страницы составляет 2 или 4 Кбайт.

Каждая операционная система поддерживает свой собственный метод работы с таблицей страниц. Обычно за каждым процессом, находящимся в основной памяти, закреплена отдельная таблица страниц. В этом случае указатель на таблицу страниц хранится в РСВ соответствующего процесса.

Аппаратная поддержка страничной организации памяти. Преобразование логического адреса в физические осуществляется для каждого адреса, генерируемого процессором, поэтому часто для ускорения этого процесса применяются аппаратные методы, например ассоциативные регистры (associative registers).

Каждый ассоциативный регистр, кроме операций чтения-записи, может обрабатывать операцию сравнения кода, поступающего на его вход с частью кода, хранимого в регистре. Матрица ассоциативных регистров хранит часть таблицы страниц. Номер страницы N подается одновременно на входы всех ассоциативных регистров, которые параллельно выполняют операцию сравнения. На выходе матрицы ассоциативных регистров образуется начальный адрес страничной рамки f того регистра, в котором произошло совпадение кода.

В том случае, если требуемый номер страницы находится в таблице страниц, т. е. ни в одном из ассоциативных регистров не произошло совпадения, происходит обращение к таблице страниц, находится искомый номер страничной рамки, а найденная строка таблицы страниц переписывается в один из ассоциативных регистров.

Защита страничной памяти основана на контроле уровня доступа к каждой странице, возможны следующие уровни доступа:

- только чтение;
- чтение и запись;
- только выполнение.

В этом случае каждая страница снабжается 3-битовым кодом уровня доступа. При трансформации логического адреса в физический сравнивается значение кода разрешенного уровня доступа с фактически требуемым. При их несовпадении работа программы прерывается.

Управление виртуальной памятью

Все методы управления памятью имеют одну и ту же цель — хранить в памяти мультипрограммную смесь, необходимую для мультипрограммирования. Рассмотренные ранее методы предполагали, что вся программа перед выполнением должна быть размещена в основной памяти. *Виртуальная память* — это технология, которая позволяет выполнять процесс, который может только частично располагаться в основной памяти. Таким образом, виртуальная память позволяет выполнять программы, размеры которых превышают размеры физического адресного пространства.

Перемещение страниц по запросу (*demand paging*). Виртуальная память чаще всего реализуется на базе страничной организации памяти, совмещенной со свопингом страниц.

Свопингу подвергаются только те страницы, которые необходимы процессору. Таким образом, перемещение страниц по запросу означает:

- программа может выполняться CPU, когда часть страниц находится в основной памяти, а часть — во внешней;
- в процессе выполнения новая страница не перемещается в основную память до тех пор, пока в ней не возникла необходимость.

Для учета распределения страниц между внешней и основной памятью каждая строка таблицы страниц дополняется битом местонахождения страницы (*valid/invalid bit*).

В том случае, если процессор пытается использовать страницу, помеченную значением *invalid*, возникает событие, называемое *страничная недостаточность* (*paging fault*).

Страничная недостаточность вызывает прерывание выполнения программы и передачу управления операционной системе. Реакция операционной системы на страничную недостаточность заключается в том, что необходимая страница загружается в основную память.

Основные этапы обработки страничной недостаточности приведена на рис. 1.24.

1. Процессор, прежде чем осуществлять преобразование логического адреса в физический, проверяет значение бита местонахождения необходимой страницы.

2. Если значение бита *invalid*, то процесс прерывается и управление передается операционной системе для обработки события страничная недостаточность.

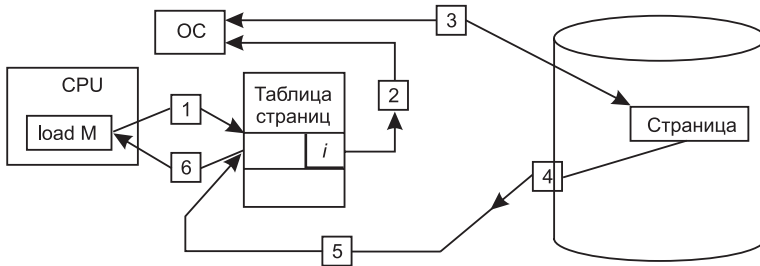


Рис. 1.24. Обработка страничной недостаточности

3. Отыскивается необходимая страница во вторичной памяти и свободная страничная рамка в основной.

4. Требуемая страница загружается в выбранную страничную рамку.

5. После завершения операции загрузки редактируется соответствующая строка таблицы страниц, в которую вносится базовый адрес и значение бита местонахождения — *valid*.

6. Управление передается прерванному процессу.

Метод обмена страниц по запросу позволяет начать выполнение процесса даже в том случае, когда ни одна страница этого процесса не загружена в основную память.

Вторичная память, используемая при обмене страниц по запросу, — это высокоскоростное дисковое устройство, часто называемое оборудованием свопинга (*swar device*), а часть используемого дискового пространства — пространство свопинга (*swar space*).

Замещение страниц. В процессе обработки страничной недостаточности операционная система может обнаружить, что все страничные рамки основной памяти заняты и, следовательно, невозможно загрузить требуемую страницу. В этом случае возможны следующие режимы: приостановка прерванного процесса, уменьшение на единицу количества процессов мультипрограммной смеси для освобождения всех ею занимаемых страничных рамок, использование метода замещения страниц.

Метод замещения страниц состоит в том, что в основной памяти выбирается наименее важная/используемая страница, называется *страница-жертва* (*victim page*), которая временно перемещается в пространство свопинга, а на ее место загружается страница, вызываемая страничной недостаточностью.

Обработка страничной недостаточности с учетом замещения осуществляется по следующему алгоритму:

- определяется местонахождение страницы путем анализа бита местонахождения;
- если значение бита *invalid*, то разыскивается свободная страничная рамка;
- если имеется свободная страничная рамка, то она используется;
- если свободной страничной рамки нет, то используется алгоритм замещения, который выбирает страницу-жертву;
- страница-жертва перемещается в пространство свопинга и таблица страниц редактируется;
- требуемая страница загружается на место страницы-жертвы и соответствующим образом редактируется таблица страниц.

Управление передается прерванному процессу. Приведенный алгоритм замещения требует двухстраничных перемещений:

- страница-жертва перемещается в пространство свопинга;
- требуемая страница перемещается в освободившуюся страничную рамку.

Страницу-жертву можно не копировать в пространство свопинга в том случае, если за время, прошедшее от последнего перемещения, ее содержимое не модифицировалось. В этом случае время замещения уменьшается примерно вдвое.

Для учета факта модификации страницы в таблицу страниц вводится дополнительный бит, который меняет свое значение на противоположное в том случае, если содержимое страницы изменилось.

Для практического использования метода обмена страниц по запросу необходимы два алгоритма:

- распределения страничных рамок (*frame allocation algorithm*);
- замещения страниц (*page replacement algorithm*).

Алгоритм распределения страничных рамок

Алгоритм распределения страничных рамок решает, сколько страничных рамок в основной памяти выделить каждому из процессов мультипрограммной смеси. Алгоритм замещения страниц решает, какую из страниц выбрать в качестве жертвы.

FIFO (first in first out). Наиболее простым алгоритмом замещения страниц является алгоритм FIFO. Этот алгоритм ассоциирует с каждой страницей время, когда эта страница была помещена в память. Для замещения выбирается наиболее старая страница.

Учет времени необязателен, когда все страницы в памяти связаны в FIFO-очередь, а каждая помещаемая в память страница добавляется в хвост очереди.

Алгоритм учитывает только время нахождения страницы в памяти, но не учитывает используемость страницы. Например, первые страницы программы могут содержать переменные, используемые на протяжении работы всей программы. Это приводит к немедленному возвращению к только что замещенной странице.

Оптимальный алгоритм. Этот алгоритм имеет наилучшее соотношение количества замещенных страниц к количеству ссылок. Алгоритм строится по следующему принципу: замещается та страница, на которую нет ссылки на протяжении наиболее длительного периода времени. Для реализации этого алгоритма необходимо каждый раз сканировать весь поток ссылок, поэтому он нереализуем на практике и используется для оценки реально работающих алгоритмов.

Алгоритм LRU (least recently used). Алгоритм выбирает для замещения ту страницу, на которую не было ссылки на протяжении наиболее длинного периода времени. Он ассоциирует с каждой страницей время последнего использования этой страницы. Для замещения выбирается та страница, которая дольше всех не использовалась. Обычно применяются два подхода при внедрении этого алгоритма:

- *подход на основе логических часов (счетчика)* — ассоциируют с каждой строкой таблицы поле «время использования», а в CPU добавляются логические часы. Логические часы увеличивают свое значение при каждом обращении к памяти. Каждый раз, когда осуществляется ссылка на страницу, значение регистра логических часов копируется в поле «время использования». Заменяется страница с наименьшим значением в отмеченном поле путем сканирования всей таблицы страниц. Сканирование отсутствует при использовании подхода на основе стека;
- *подход на основе стека номеров страниц* — стек номеров страниц хранит номера страниц, упорядоченных в соответ-

ствии с историей их использования, *на «вершине» стека* располагается только что использованная страница, а *на «дне»* дольше всех неиспользуемая страница. Как только осуществляется ссылка на страницу, она перемещается на вершину стека, а номера всех страниц сдвигаются вниз.

1.4. Связь с оператором

В общем случае, конечно, следует говорить о *связи с внешней средой*, поскольку, например, при использовании ЭВМ в системах управления технологическими комплексами (производство, летательные аппараты, корабли и пр.) человек может быть исключен (полностью или частично) из контура управления и внешними устройствами ЭВМ будут *датчики* (скорости, высоты, давления, температуры) и *эффе́кторы* (приводы рулей, манипуляторы, сервомоторы клапанов и пр.).

Связь с пользователем

Связь с пользователем, сокращенно поименованная здесь как связь с оператором, — как говорят англичане, *last but not least* — последняя в списке, но не по важности функция ОС.

Типология связи с человеком определяется как уровнем развития программного обеспечения, так и техническими средствами. Как это следует из рис. 1.1, связь с пользователем включает:

- командный (или иной) интерфейс по управлению системными процессами в вычислительной системе (собственно функции оператора ОС). Пользователь (привилегированный) осуществляет запуск-останов программ, подключение-отключение устройств и прочие релевантные операции;
- интерфейс по управлению пользовательскими процессами (контроль состояния процесса, ввод-вывод данных в процесс/из процесса).

В состав *пользователей* в общем случае включаются следующие группы лиц, контактирующих с системой:

- администратор системы — лицо или группа, отвечающая за сопровождение данных, назначение уровней доступа, включение/исключение пользователей;

- оператор системы, осуществляющий сопровождение вычислительного процесса;
- прочие пользователи (не обладающие привилегиями доступа к данным), в том числе:
 - операторы подготовки данных (ОПД) — персонал, осуществляющий ввод данных с рабочих листов или документов на основе соответствующих инструкций, в среде специальных программных интерфейсов;
 - интерактивные пользователи (ИП) — лица, имеющие доступ на ввод, коррекцию, обновление, уничтожение и чтение данных в рамках, как правило, ограниченной области БД;
 - конечные пользователи (КП) — лица, использующие БД для получения справок и решения задач.

Очевидно, что именно *оператор ЭВМ является естественным пользователем ОС*; все же прочие пользователи становятся таковыми лишь вследствие расширения функций пользователя в связи с интеграцией (особенно в случае персональных ЭВМ) функций конечного пользователя, администратора системы и оператора.

Интерфейс — это способ общения пользователя с персональным компьютером, пользователя с прикладными программами и программ между собой. Интерфейс служит для удобства управления программным обеспечением компьютера. Интерфейсы бывают однозадачные и многозадачные, однопользовательские и многопользовательские. Интерфейсы различаются между собой по удобству управления программным обеспечением, т. е. по способу запуска программ. Существуют универсальные интерфейсы, допускающие все способы запуска программ, например Windows 3.1, Windows 95/98/ME/NT/XP.

Разновидности интерфейсов

Интерфейсы различаются по способу доступа к командным файлам программ.

Командный (текстовый) интерфейс. Всякая операционная система имеет командный интерфейс (иногда в скрытой форме).

Если снять «шелуху» текстовых или графических оболочек или интерфейсов, то «на глубине» вы *всегда* найдете командный интерфейс.

В первой из ОС (OS/360) взаимодействие с пользователями было жестко поделено между следующими компонентами:

- *командный язык оператора ЭВМ* (лицо, ответственное за управление вычислительным процессом). Это *язык диалогового режима* — команды запуска-остановки задач, привязки носителей информации к устройствам, получения информации о заданиях, ожидающих выполнения, вывода, наличия свободной памяти и свободных устройств и др.;
- *язык управления заданиями* (JCL — Job Control Language), на котором прочие пользователи (программисты, разработчики и просто конечные пользователи) описывали состав и структуру процесса обработки данных — последовательность запуска программ, входные и выходные файлы, условия, при которых те или иные программы должны быть выполнены или пропущены, и др. Это язык пакетной обработки, не допускающий вмешательства пользователя в собственно процесс вычислений, компиляции и пр.

По мере развития ЭВМ, ОС, появления и широкого распространения диалоговых устройств (видеотерминалов) в последующих ОС произошла интеграция данных компонент в единый командный язык. Для разграничения между командами оператора, администратора, конечного пользователя используются методы разделения доступа и назначения привилегий, в то время как формат команд является достаточно единообразным.

Далее, после распространения ПЭВМ данное разграничение сошло на нет (в ОС MS DOS), поскольку пользователь ПК в едином лице соединяет функции оператора, администратора, конечного пользователя. Затем с появлением локальных сетей и более мощных ПК, работающих в многопользовательских режимах, в сетевых ОС и ОС ПЭВМ, вновь организуется разграничение доступа и т. д. Таким образом, данный процесс является циклическим (точнее, спиралевидным).

В табл. 1.3 приведена выборка из основных функциональных групп команд различных ОС.

В большинстве ОС в настоящее время сложился более или менее унифицированный *формат командной строки*. Командная строка включает в себя (рис. 1.25, табл. 1.4):

- тип операции (имя команды или программы);
- рабочий вход (входные файлы или устройства);
- рабочий выход (выходные файлы или устройства);

Таблица 1.3. Сравнительный анализ некоторых команд различных ОС

Команда (функция)	OS/360/370	RSX-11/20	MS DOS	Unix
1. Спецификация устройств, файлов и пр. в разных ОС				
Обозначения устройств (НМД)	UNIT=SYSDA, UNIT=191, 192	DKx: DPx: . . .	A:, B:, C:	ROOT/DEV/SD A
Устройства печати	SYSOUT=A	LPx:	LPTx:	lpr, lp
Терминалы	UNIT=050, 0C0	TTx:	CON	TTYx:
Спецификация файла	DSN, VOL, DCB, SPACE	Устр.: [G, N] имя.тип; версия	Устр.: \путь\ имя.тип	/путь/ имя.тип
Спецификация группы файлов	Партиционный файл (НД) (DSORG=PO) Сцепленные НД	Использование масок «*», «?»	Использование масок «*», «?»	Использование масок «*», «?»
2. Подготовка носителей к использованию, работа с каталогами				
Инициализация диска, создание файловых систем	INIT	INI, FMT	FDISK, FORMAT	mkfs
Создание каталогов в ФС	DISP=NEW	UFD	MKDIR	mkdir
Проверка диска, файловой системы	DASDR/CHK	BAD/CHK	CHKDISK	fchk
Проверка/установка файловых атрибутов	DISP=SHR, OLD, NEW	—	ATTRIB	file, chmod
Удаление каталога	DISP=DELETE	PIP/DE	RD	rmdir
3. Работа с файлами				
Копировать файл	DISP=NEW, DSNANE=NEWFILE	PIP A:=B	COPY	cp
Переместить файл	DISP= (OLD, DELETE)	PIP A:=B	MOVE	mv
Переименовать файл	DISP=NEW, DSN=NEWFILE	PIP A:=B	REN, RENAME	mv
Удалить файл	DISP=DELETE	PIP/DE	DEL, ERASE	rm

Окончание табл. 1.3

Команда (функция)	OS/360/370	RSX-11/20	MS DOS	Unix
Поиск в файле по контексту	—	—	FIND	grep, awk
Архивация файла на МД или МЛ	DASDR/DUMP-RESTORE	BRU (Backup-restore utility)	BACKUP, RESTORE	TAR
Распечатка содержимого файла	IEBGGENER-PRINT	PIP LP:=...	TYPE, COPY	cat
Сравнение файлов	—	—	FC	comm (Common) cmp (Compare) diff (difference)
4. Некоторые информационные команды				
Текущий диск, каталог	D U (используемые устройства)	UIC	PATH	pwd
Содержание диска, каталога	VTOC	PIP/LI	DIR	li
Активные задачи, пользователи	DA (Display Active)	—	—	who
5. Некоторые команды управления процессами				
Начало работы пользователя (задания)	//AAA JOB	HEL (hello)	—	login
Окончание работы пользователя	//	BYE	—	<Ctrl+D>, ^D
Запуск последовательности программ	JCL-операторы TSO-команды	Командный файл .CMD	Пакетный файл .BAT	Пакетный файл Shell
Запуск задачи, программы, процесса	S (Start) E (Exec)	RUN	Имя файла (пакетного .BAT или исполняемого .COM, .EXE)	Имя файла
Остановка программы	C (Cancel) P (Stop)	ABO (Abort)	<Ctrl+C>, <Ctrl+Break>, ^C	^C

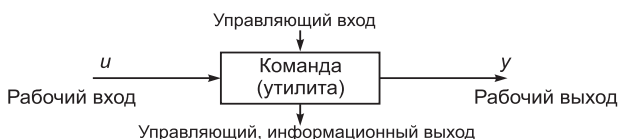


Рис. 1.25. Типовая структура командной строки языков операционных систем

Таблица 1.4. Примеры командных строк и их форматов

Команда	Рабочий вход	Рабочий выход	Управляющий вход	Информационный выход
OC MS DOS copy \test.txt a: >prn	Файл test.txt	Копия файла на дисковом a:	Режим по умолчанию	Протокол на устройстве печати (prn)
OC MS DOS format a: /v >prn	Исходная дискета на устройстве a:	Форматированная дискета a:	Ключ команды /v (запросить метку диска)	Протокол форматирования на устройстве prn
OC RSX BCD LP:=MT00	Магнитная лента на устройстве MT0:	Нет	Нет (режим по умолчанию)	Содержание ленты на устройстве LP: (печать)
OC Unix ls -l /bin	Каталог /bin	Нет	Ключ -l (распечатка в полном формате)	Содержание каталога на экране терминала

- управляющий вход (управляющие параметры или ключи команды);
- управляющий выход (обычно — протокол, содержащий диагностику ошибок, код завершения или другую информацию).

Текстовый или графический полноэкранный интерфейс. Он имеет, как правило, в верхней части экрана систему меню с подсказками. Меню часто бывает выпадающим (ниспадающим — pull-down).

Для управления компьютером курсор экрана или курсор мыши после поиска в дереве каталогов устанавливается на командные файлы программ (*.exe, *.com, *.bat) и для запуска программы нажимается клавиша <Enter> или правая кнопка мыши. Различные файлы могут выделяться разным цветом или иметь разный рисунок. Каталоги (папки) отличаются от файлов размером или рисунком. Данный интерфейс является основным для всех видов программных оболочек. Пример: Norton Commander и «нортонообразные» оболочки (DOS Navigator, Windows Commander, Disk Commander). Подобный интерфейс имеют инструменты Windows 3.1 (Диспетчер файлов) и Windows 95/XP

(Мой компьютер и Проводник). Такой интерфейс весьма удобен, особенно при работе с файлами, поскольку обеспечивает высокую скорость выполнения операций, позволяет создавать пользовательские меню, запускать приложения по расширению файлов, что повышает скорость работы с программами.

Графический многооконный пиктографический интерфейс. Представляет собой рабочий стол (DeskTop), на котором располагаются пиктограммы (значки или иконки программ). Все операции производятся, как правило, мышью. Для управления компьютером курсор мыши подводит к пиктограмме и запуск программы осуществляют щелчком левой кнопки мыши по пиктограмме. Это наиболее удобный и перспективный интерфейс, особенно при работе с программами. Примеры: интерфейс компьютеров Apple Macintosh (Mac OS), Windows 3.1, Windows 95/98/2000/XP, OS/2, GNOME, KDE.

Терминалы

Терминалы, или *терминальные устройства*, ЭВМ являются важнейшей компонентой систем, основанных на человеко-машинном взаимодействии. Это *диалоговые* или *интерактивные* устройства, предназначенные для ввода-вывода небольших количеств информации, первоначально с целью *управления* вычислительным процессом и наблюдения за его ходом, а в дальнейшем — также для ввода-вывода *исходных* данных и *результатов* работы программ. Первоначально в ЭВМ использовались в качестве терминалов *механические* устройства, заимствованные из смежных технологий — связь и оргтехника, — телетайпы (типа ТА-67), телеграфные аппараты (СТА-2М), электрические пишущие машинки (ПМ типа CONSUL). Это был довольно длительный период, в течение которого сложились определенные стандарты, приемы работы оператора и протоколы ввода-вывода и интерпретации данных. Строка информации, вводимая оператором, являлась, как правило, *командой*, требующей выполнения определенных действий от ЭВМ (ОС). Конечная ширина листа (или бумажной ленты) ПМ (80 знаков) ограничивала длину возможных команд. Признаком окончания ввода команды являлось нажатие клавиши <BK> (*возврат каретки*, она же <CR> — <Carriage Return>, <Return>, <Enter> и пр.). Реакция системы (ответ на запрос, сообщение об ошибке, небольшая порция выходных данных) также выводилась строками по 80 символов,

образуя вместе с копиями команд *протокол диалогового сеанса* (или журнал — log) в бумажной форме.

Низкие скорость обмена информацией с ЭВМ и надежность механических терминалов, а также трудности с исправлением информации (*редактированием*) ограничивали применимость и, в частности, делали бессмысленным их использование пользователями-программистами для отладки программ и прочих манипуляций. В ранних версиях операционной системы OS/360 и других систем того времени единственный механический терминал устанавливался в машинном зале и предназначался для *оператора ЭВМ*. Это устройство получило название *консоль** (console). На крупных вычислительных установках их могло быть несколько (Master Console, Alternative Console и пр.).

Появление в начале 1970-х годов *электронных терминалов*, специально разработанных для использования с ЭВМ, привело к настоящему перевороту в применении машин, существенно приблизив все типы пользователей к вычислительному процессу, облегчив разработку и отладку программ, а также эксплуатацию автоматизированных систем.

Электронный или *видеотерминал* — CRT-device (Catode Ray Tube — устройство с электронно-лучевой трубкой), VDU (Video Display Unit — устройство отображения информации), первоначально получивший в отечественной практике наименование *дисплей*, представляет собой клавиатуру (keyboard), сопряженную с экранным устройством (screen). Ранние модели видеотерминалов (VT) не были избавлены от наследия ПМ — состав клавиатуры, построчный ввод и исправление ошибок, прокручивание экрана наподобие бумажной ленты (scrolling) и, самое главное, *символьный* (алфавитно-цифровой) характер выводимой информации, хотя, как это хорошо известно из опыта телевидения, никаких технических ограничений экран (в отличие от каретки ПМ) не вносит. Более совершенные VT, разработанные в 1980-е годы (IBM 3270, VT-100), во многом определили современное состояние устройств:

- появились возможности прямого доступа к информации на экране (для ввода и корректировки);

* Как известно, в строительстве и архитектуре *консолью* именуют конструкцию, состоящую из горизонтальной балки, опирающейся на подкос. Именно так выглядел столик для ПМ (*пульт оператора*), прикрепленный к *инженерному пульту* управления большой ЭВМ.

- на клавиатуре добавились *функциональные* клавиши, реакция на которые определялась программой, работающей с ВТ;
- появились *клавиши редактирования* — <Del>, <Ins>;
- появились *клавиши управления курсором* (для выбора места на экране);
- добавилась *управляющая клавиша* <Control> (<Ctrl>), модифицирующая вводимый код, при одновременном нажатии с символьной клавишей и т. п.

Однако это все еще были алфавитно-цифровые устройства, отображающие на экране массив символьной информации размером в 80 столбцов на 17 строк (т. е. до 1600 символов).

Типовая конфигурация машины (до появления ПЭВМ) включала в себя 8 (или 16, или 32) *терминалов пользователя*, размещенных в специальных помещениях (*дисплейные классы*), и одну или более *дисплей-консоли* (терминал оператора), размещенную поближе к месту основных событий (в машинном зале).

Конфигурация ПЭВМ, в которую входит единственный ВТ (монитор), является *частным* (или, как выражаются математики, *вырожденным*) случаем общей конфигурации, при этом ВТ несет бремя нагрузки как консоли, так и пользовательского терминала.

Терминал ПЭВМ (в дальнейшем будет упоминаться как *консоль*, поскольку в MS DOS это устройство обозначается как CON), в отличие от старинных ВТ, базируется на *графическом выводе* информации (в растровой форме) на экран, что дает возможность отображать не только обычную символьную информацию, но и *квазисимвольную* (элементы электронных схем, шахматные фигуры, редкие алфавиты). Наконец, на подобный ВТ может быть выведена произвольная растровая информация*.

Рассмотрим подробнее *клавиатуру и экран* консоли. *Клавиатура* (рис. 1.26) включает следующие области (заметим, что ряд областей или отдельных клавиш продублирован).

* Терминалы ПЭВМ (относящиеся к так называемым ANSI-терминалам) по мере развития технических средств претерпели ряд изменений: улучшение разрешающей способности (количество точек-пикселей на экране), увеличение числа уровней яркости и количества отображаемых оттенков цвета. Это развитие осуществлялось не столько за счет усовершенствования «телевизора», сколько путем разработки новых контроллеров (управляющих карт, или *адаптеров*). Ранние ПЭВМ были укомплектованы контроллерами CGA (Color Graphic Adapter), затем появились EGA (Enhanced Graphic Adapter), VGA (Video Graphic Array, обеспечивающий качество, близкое к видеоизображению телевизора) и т. п.

1. *Символьная область.* Здесь находятся клавиши, являющиеся основными для ПМ и механических терминалов, — строка цифровых клавиш, три строки буквенных клавиш, пробел (<Space>). Необходимость совместного использования символов латиницы (А—Z) и кириллицы (А—Я) создает проблему размещения символов по клавишам. Как известно, месторасположение символов отражает их частоту и совместную встречаемость в словах соответствующего языка, в связи с чем отечественные клавиатуры в первой символьной строке содержат буквы ЙЦУКЕН, англо-американскому стандарту соответствует строка QWERTY, континентально-европейскому стандарту — AZERTY.

Первые отечественные терминалы использовали в качестве основы размещение ЙЦУКЕН, привязывая к символам кириллицы соответствующую им по *правилам транслитерации* латиницу: Й/Ј, Ц/С, У/Ў, К/К, Е/Е, Н/Н и т. п. На консоли ПК поддерживаются два стандарта, и размещение символов имеет вид Q/Й, W/Ц, Е/У, R/К, Т/Е, Y/Н, что обычно смущает начинающего пользователя.

2. *Функциональная клавиатура (ФК),* сохранившаяся как знак преемственности со старыми терминалами, хотя принципы обмена информацией консоль—ЭВМ здесь таковы, что необходимость в ней отсутствует (*вся клавиатура является программно-управляемой*). За последние годы сложились определенные стандарты де-факто применения ФК, например <F1> — HELP (Помощь, подсказка), <F10–F12> — Quit (Завершение работы программы) и т. п.

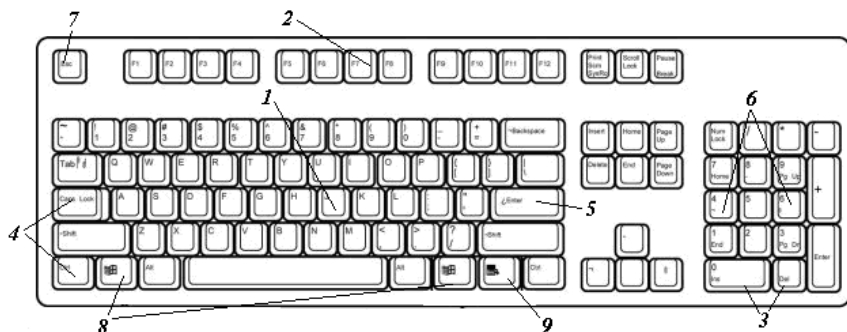


Рис. 1.26. Структура клавиатуры консоли:

- 1 — символьная область; 2 — ФК; 3 — клавиши редактирования; 4 — управляющие клавиши; 5 — ввод; 6 — управление курсором; 7 — <Escape>; 8 — клавиши <Windows>; 9 — клавиша <Application>

3. *Клавиши редактирования* — <Ins> — включение/выключение режима вставки символов, <Del> — удаление текущего символа, <BS>, <BackSpace> — удаление символа слева.

4. *Управляющие клавиши* (изменяют значение нажимаемого одновременно с ними символа):

<Shift> — переключение регистров, имеется также на любой ПМ. В буквенной области <Shift> переключает строчные символы на прописные, в цифровой области — цифры на служебные символы (@ # \$ % ^ и т. п.);

<CapsLock> — фиксация верхнего регистра, в отличие от ПМ, действует только на буквенные клавиши;

<Ctrl> — появляется впервые на VT100. Позволяет вводить коды, которым не соответствуют какие-либо обычные символы. Например, <Ctrl+Z> вводит символ EOF — *конец файла*;

<Alt> — появляется на ANSI-терминале. Расширяет возможности <Ctrl>. Например, <Alt+2+1+9> вводит █ — символ так называемой *псевдографики*.

5. <Enter> — *ввод*. Является символом окончания строки, соответствует клавише <BK> механического терминала, клавиша продублирована.

6. *Клавиши управления курсором*: <←> — стрелка влево, <→> — стрелка вправо, <↑> — стрелка вверх, <↓> — стрелка вниз, <Home> — начало, <End> — конец, <PgUp> — страница назад, <PgDn> — страница вперед. Действие клавиатуры рассматривается ниже. Клавиши продублированы. Основная зона расположена на правом краю клавиатуры и совмещена со второй цифровой клавиатурой (основная размещена в символьной области). Переключение регистров на этой клавиатуре (цифры или управление курсором) осуществляется клавишей <NumLock> — зафиксировать цифровой режим. Дубль клавишей управления курсором находится левее, перед символьной областью.

7. Клавиша <Escape> (Выйти) впервые появляется на VT100 и реализует выход из текущей программы. Обычно так же программируется и на ПЭВМ.

В заключение разговора о клавиатуре поясним, что понималось выше под *программируемостью*. Это означает, что интерпретация всех перечисленных клавиш не обязательно соответствует тем или иным символам/действиям, которые на них обозначены. *Нажатие на клавишу вырабатывает не код символа, а номер клавиши* (поэтому основные и дублирующие символы/области в принципе различаемы). Эта информация затем обрабатывается *драйвером*

клавиатуры — программой, постоянно находящейся в ОП и преобразовывающей номер клавиши в код символа, который выводится на экран и поступает в распоряжение работающей прикладной программы. Этот же драйвер ответственен за переход с *латиницы на кириллицу*. В отличие от механических терминалов и старинных ВТ, на консоли нет клавиши переключения <Лат/Рус>. Поскольку драйверов весьма много (всякий себя уважающий программист в середине своей карьеры, как правило, пишет свой драйвер), надо иметь в виду, что переключение может осуществляться по-разному. Обычно используются сочетания управляющих клавиш (например <Shift+Alt>, <Shift+Shift> — левая и правая клавиши и пр.). Отсутствие лампочки, которая на старых ВТ указывала, что включен регистр «Лат» (или «Рус»), компенсируется обычно появлением на экране цветовой рамки или надписей LAT, RUS (или EN/RU) в углу экрана.

Очевидно, драйвер управляет привязкой символов к клавишам. Обычно размещение букв не вызывает проблем — это стандарты QWERTY и ЙЦУКЕН. Размещение же служебных символов (! @ # \$ % ^) и пр. может различаться в разных драйверах и может не соответствовать в связи с этим символам, нанесенным на клавиши. И с буквами не все однозначно, особенно достается буквам Ъ и Ё (иногда даже Ъ и ъ могут оказаться на разных клавишах).

Экран

За последние десятилетия сформировались следующие основные режимы представления и управления информацией на экране, которым соответствуют определенные сценарии диалога человек—ЭВМ в текстовом режиме:

- режим командной строки;
- режим форматированного экрана;
- режим меню.

Режим командной строки. Работа в этом режиме мало чем отличается от работы с механическим терминалом, или с ранними ВТ.

Экран состоит из двух областей — *командная строка* (command line, current line), аналог строки, на которой находится каретка ПМ, и *протокол диалога* (history), аналог отрезка бумажной ленты. На экране (рис. 1.27) пользователь ввел команду dir,

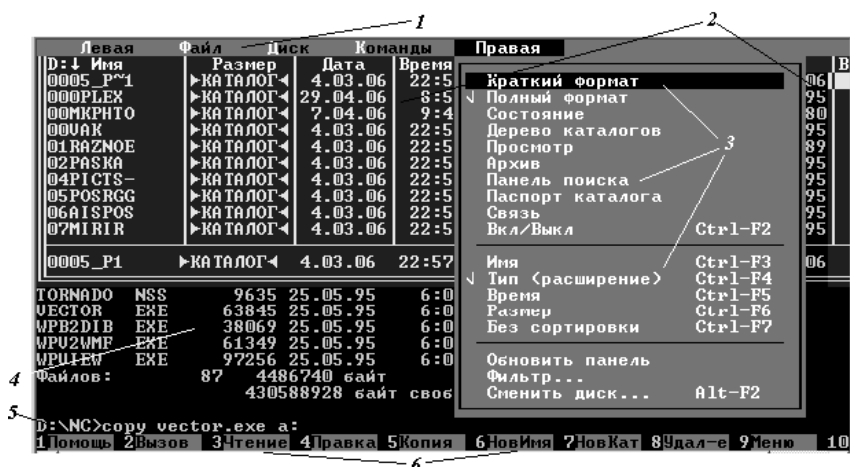


Рис. 1.27. Экран Norton Commander — использование различных элементов интерфейса:

1 — ниспадающее меню; 2 — левая и правая панели (всплывающие меню); 3 — вертикальное меню; 4 — видимый участок протокола (команда dir); 5 — командная строка; 6 — комментарий к ФК (горизонтальное меню)

просмотрел результаты и вводит команду `copy`. Однако здесь появляется важный новый объект (как и на старинных ВТ) — *курсор*, или *активная область экрана*, с которой пользователь работает в данный момент. В большинстве ситуаций (и в данном режиме тоже) эта область занимает минимально доступную площадь — *одно знакоместо*, т. е. площадь, на которой для данного терминала отображается один *знак* (символ). Попытки перевести слово *cursor* как *бегунок* и т. п. не прижились. Курсор выделяется яркостью, мерцанием или цветом, так что его всегда можно распознать. При вводе символ в строке появляется в позиции курсора, а тот автоматически смещается вправо.

При необходимости откорректировать строку используют клавиши `<←>` и `<→>`, перемещающие курсор к месту исправления. Остальные клавиши управления курсором в командном режиме не задействованы. Из этого правила, конечно, есть исключения — в некоторых системах клавиша `<↑>` позволяет вызвать в командную строку предшествующую команду, что создает известные удобства при вводе серии сходных команд. Клавиши `<↑>` и `<↓>` здесь дают возможность *перелистать* журнал, где хранятся введенные команды, выбрать, исправить и выполнить требуемую команду.

Режим форматированного экрана (ФЭ) — рис. 1.28. В этом случае экран представляет собой совокупность *окон*, каждое из которых содержит некоторое элементарное данное и обычно снабжено текстовым комментарием (как правило, название данного). Если командный режим в основном ориентирован на управление вычислительными процессами (хотя и может быть применен для *ввода-вывода данных*), ФЭ — в основном именно на эти функции (здесь, как и везде, есть исключения, например, какое-то из окон может использоваться для ввода команд).

Традиционно основным способом использования ФЭ является работа с *файлами данных*, или совокупностями *агрегатов данных (записей)* одинаковой структуры.

Видимая на экране в режиме ФЭ запись является так называемой *текущей* или *активной*. Данные, составляющие содержимое ее полей, находятся в оперативной памяти и могут быть изменены путем подвода курсора и редактирования в окне. Интерпретация клавиш управления курсоров в ФЭ зависит от программы, с которой осуществляется работа, однако сложились следующие стереотипы, справедливые для многих популярных программных средств:

<←>, <→> — переход внутри окна на одну позицию;

<↑>, <↓> — переход к следующему/предшествующему окну;

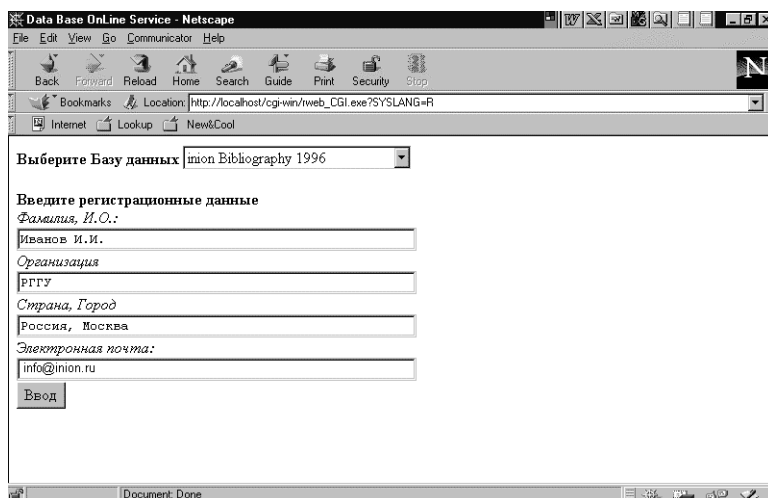


Рис. 1.28. Пример форматированного экрана (окна для ввода данных пользователем)

<Home> — переход к началу поля;
<End> — переход к концу поля;
<PgUp> — вызвать предшествующую запись (переместить указатель текущей записи к началу файла);
<PgDn> — вызвать последующую запись (переместиться к концу файла).

Примечания:

1. В окне может быть представлена только часть соответствующего поля. Если таковое имеет слишком большую длину, тогда осуществляется *горизонтальная прокрутка* (scrolling) содержимого при нажатии клавиш <←>, <→>, <Home>, <End>.

2. Запись большого размера может занимать несколько экранов, и тогда клавиши <PgUp>, <PgDn> вначале перелистывают экраны текущей записи, а затем вызывают соседнюю запись.

3. Нажатие клавиш <←>, <→> в начале (конце) поля (окна), как правило, вызывает переход к предшествующему/последующему окну. Клавиши <↑>, <↓>, нажатые в первом (последнем) окне экрана вызывают переход к предшествующей/последующей записи.

Важным частным случаем ФЭ является *окно во весь экран*, характерное для *текстовых редакторов*, программных продуктов, предназначенных для манипуляции с *текстовыми файлами*.

Режим меню. В данном случае основными элементами экрана являются (рис. 1.29):

- падающее (ниспадающее) меню (включает горизонтальную и вертикальную части);
- всплывающее меню (обычно пункты меню расположены вертикально);
- строка состояния.

Устоявшаяся интерпретация основных клавиш здесь выглядит следующим образом:

<←>, <→> — переход внутри горизонтальной части падающего меню на одну позицию;

<↑>, <↓> — переход к следующему/предшествующему пункту вертикальной части падающего или всплывающего меню;

<Home> — переход к началу горизонтальной части;

<End> — переход к концу горизонтальной части;

<PgUp> — переход на верх вертикальной части;

<PgDn> — переход в низ вертикальной части;

<Enter> — выход из меню с выбором соответствующего пункта;

<Esc> — выход из меню без фиксации выбора.

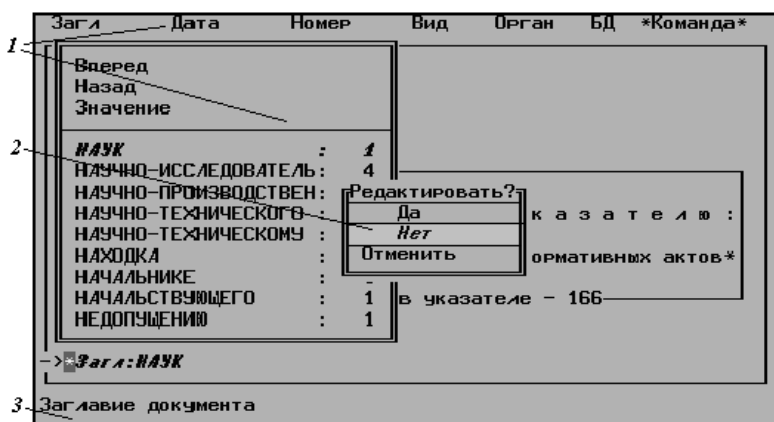


Рис. 1.29. Элементы экрана в режиме меню:

1 — ниспадающее (pull-down) меню; 2 — всплывающее (pop-up) меню; 3 — строка состояния (status bar)

Графический интерфейс пользователя (GUI — Graphics User Interface). Появление операционных систем и оболочек с развитыми диалоговыми графическими средствами (OS Macintosh, Windows 3.1, а особенно Windows 95/98/ME, а также NT/2000) и средств программирования, позволяющих создавать графические интерфейсы (FoxPro for Windows и пр.), в частности — объектно-ориентированных систем программирования, привело к внедрению и широкому распространению элементов экранного интерфейса.

Графические интерфейсы иногда обозначают следующей аббревиатурой — *WIMPD* (*Windows, Menu, Pointing Device*) — окна, меню, указывающее устройство, как основные действующие элементы в подобном интерфейсе.

Оболочка Microsoft Windows не была изначально полноценной операционной системой, так как она существует «поверх» системы MS DOS. Она возникла в виде стандартизатора графического интерфейса и прижилась исключительно потому, что пользователь хотел видеть программу, с которой ему часто приходится работать, красивой, практичной, удобной и легкой в освоении и использовании.

Для ОС Unix также был создан специальный графический интерфейс — X Window; фирма IBM выпустила вместе с операционной системой OS/2 свой вариант графического интерфейса пользователя — Presentation Manager.

Функции, используемые программой пользователя при работе с графическим пользовательским интерфейсом, схожи, как и сами интерфейсы.

Операционная система (оболочка), ориентированная на графический интерфейс пользователя, предоставляет не только функции, поддерживающие ввод-вывод, но и широкий спектр системных вызовов, позволяющих использовать различные графические примитивы: от самых простых (точки, линии, дуги) до самых сложных (области, окна, курсоры). Основным преимуществом использования графического интерфейса операционной системы является то, что с помощью него программа может создавать графические изображения, которые будут выглядеть одинаково на всех устройствах, поддерживаемых операционной системой (принцип WYSIWYG — What You See Is What You Get — что видим, то и получаем).

Большое внимание в графическом интерфейсе операционной системы обычно уделяется шрифтам. Исторически сложилось так, что первыми и долгое время единственными шрифтами для компьютеров оставались растровые (точечно-матричные) шрифты. Такие шрифты занимали малый объем памяти, однако их символы невозможно было вращать, наклонять, уменьшать без искажений или увеличивать можно было только в целое число раз. С появлением графического интерфейса операционные системы стали предоставлять системные средства для поддержки использования векторных шрифтов, которые не только легко масштабируются, меняют наклон и толщину, но и выглядят одинаково на всех устройствах, поддерживаемых операционной системой. Каждая операционная система поддерживает свой стандарт векторных шрифтов (TrueType для Microsoft Windows; Adobe Type Manager для OS/2; GhostScript для Linux).

Графический интерфейс включает следующие понятия — рабочий стол, окна, пиктограммы, элементы графического интерфейса (виджеты), указывающее устройство (мышь).

После запуска программа обычно создает окно, с которым она ассоциируется и работает. Пользователь, работая с окном и находящимися в нем объектами, заставляет операционную систему (или программную оболочку) посылать программе сообщения, активизирующие необходимые пользователю возможности программы. В процессе работы программа также может создавать другие окна (выбора, диалога, обрабатываемого файла и др.)

и получать от них сообщения, таким образом, стандартизируются часто используемые элементы диалога с пользователем.

При уменьшении некоторого окна до пиктограммы освобождается место для другого окна, которое может быть увеличено или уменьшено в соответствии с потребностями.

Если продолжить аналогию с поверхностью рабочего стола, то каждое работающее приложение можно рассматривать как, например, скоросшиватель с бумагами по определенной теме. Раскрывая скоросшиватель не полностью, можно получить возможность работать с бумагами, одновременно не теряя возможности наблюдать за ситуацией на столе. «Распахнув» скоросшиватель в полный формат, получите возможность «с комфортом» работать над содержащимися в нем бумагами, но они при этом занимают всю поверхность стола, накрывая все остальное. Завершив сегодня работу с данным скоросшивателем, можно свернуть все бумаги и закрыть скоросшиватель, не убирая его, однако, с поверхности стола.

Представление и расположение окон в значительной мере зависят от того, сколько приложений одновременно выполняется в среде. Если активно всего одно приложение, то целесообразно представить соответствующее окно в полноэкранном варианте. Работа одновременно с двумя приложениями предполагает наличие двух окон нормального размера, размещенных на двух половинах экрана, верхней и нижней (или левой и правой). При работе с большим числом приложений удобно часть приложений, в которых в данный момент пользователь не испытывает острой необходимости, представить пиктограммами.

Интерфейс оболочки представляет собой набор наглядных и естественным образом организованных средств управления приложениями. Работая в графической среде, пользователь уже для вызова приложений не вводит имена и директивы с клавиатуры, а оперирует с соответствующими пиктограммами с помощью мыши.

Графические оболочки делают технологию работы с компьютером более естественной и ясной. Большую роль здесь играет мышь как основной инструмент управления машиной. В целом ряде случаев для вызова некоторых (довольно сложных) операций достаточно просто *«перетащить и положить»* (*Drag-and-Drop*) пиктограмму или другой объект с помощью мыши. Например, в оболочке Windows 3.1 для распечатки некоторого документа достаточно с помощью мыши «вытащить» из окна Менеджера Фай-

лов (File Manager) пиктограмму соответствующего файла и «положить» ее поверх пиктограммы Менеджера Печати. Перетаскивание пиктограммы файла документа в открытое окно редактора Write загружает соответствующий документ в окно.

Основные элементы графических интерфейсов (виджеты, widgets)

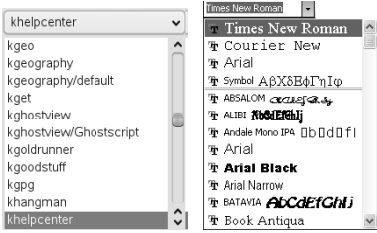
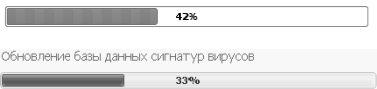
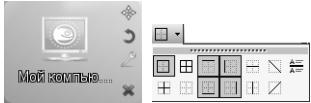
Виджет — это заготовка части пользовательского интерфейса (кнопка, часть меню, пиктограмма и т. д.) с параметрами, вызываемая к окну экрана.

Основные элементы диалогового окна, создаваемого для взаимодействия с активным приложением, приведены в табл. 1.5. Здесь использованы примеры из интерфейсов таких систем, как Windows и Linux.

Таблица 1.5. Основные виджеты графических интерфейсов

Наименование	Функция	Типичный вид
Кнопка (Button)	Предназначается для немедленного выполнения обозначенного действия	
Зависимые переключатели (радиокнопки, RadioButton)	Выбирает один из взаимоисключающих режимов работы (разновидности действий)	
Независимые переключатели (флажок, CheckBox)	Задаёт несколько взаимодополняющих атрибутов режима работы	
Дерево	Панель навигации, позволяющая управлять поиском и просмотром альтернатив	

Окончание табл. 1.5

Наименование	Функция	Типичный вид
Раскрывающийся список (List)	Содержит все возможные (взаимоисключающие) значения атрибута, из которых пользователь должен выбрать одно	
Стрелки навигации	Позволяют перемещаться по запомненному списку страниц (узлов)	
Полоса прокрутки (ScrollBar)	Осуществляет управление просмотром открытого окна	
Индикатор выполнения	Отражает состояние выполнения некоторого процесса/программы	
Регулятор/ползунок	Устанавливает регулируемый параметр в интервале от минимума до максимума	
Контекстное меню	Устанавливает перечень действий, связанных с определенным узлом навигации (ярлыком, иконкой, кнопкой и пр.)	
Заголовок окна	Содержит наименование приложения/документа и кнопки управления размером окна	
Вкладки	Параметры сгруппированы на двух и более вкладках. Выбранная вкладка отображается впереди остальных	

Естественно, графические интерфейсы предусматривают также включение и использование ранее перечисленных элементов интерфейса — всплывающие (PopUp) и свешивающиеся/ниспадающие (PullDown) меню, строки состояния (StatusBar), окна ввода и окна командной строки (Terminal) — рис 1.30. Просьба к читателям самостоятельно идентифицировать на данном экране как перечисленные выше, так и иные (например, гиперссылки) элементы интерфейса.

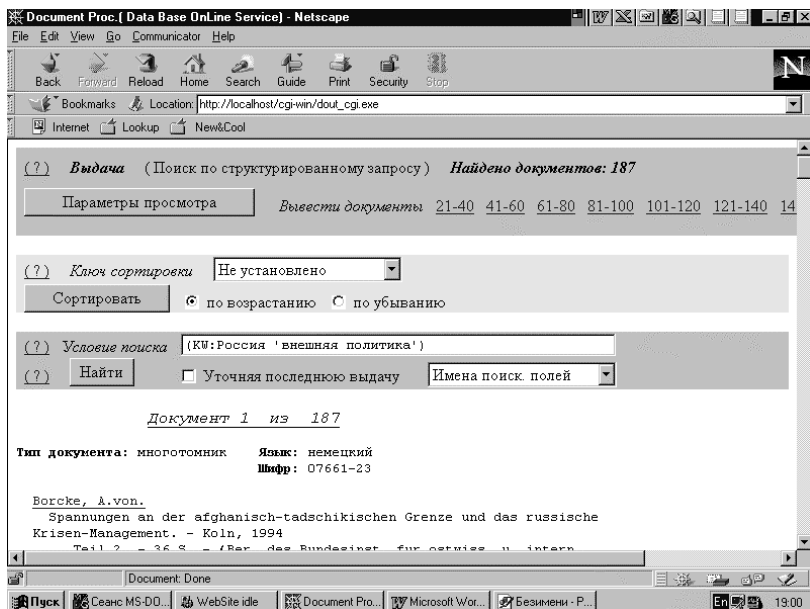


Рис. 1.30. Комбинация элементов интерфейса в реальном приложении WinIrbis

Кроме того, при рассмотрении конкретных систем и связанных с ними приложений будет описан и ряд других виджетов, и их типичных комбинаций.

Контрольные вопросы

1. Каков состав программного обеспечения ЭВМ?
2. В чем заключаются функции операционных систем?
3. В чем заключается управление данными?

4. Какова организация файлов на МЛ?
5. Что такое логическое начало ленты? Физическое начало ленты?
6. Для чего используется блокирование записей на магнитных носителях?
7. Приведите аргументы за и против увеличения длины блока на МЛ (МД).
8. Какие виды нарушения целостности данных в простейшей ФС вы можете привести?
9. Изобразите схематическую структуру ФС с косвенной адресацией и приведите примеры нарушения целостности данных.
10. Охарактеризуйте основные типы файлов. Дайте перечень основных типов текстовых файлов.
11. Дайте основные характеристики накопителей на гибких магнитных дисках.
12. В чем заключается управление задачами в ОС?
13. Что такое процесс в вычислительной системе? Какие типы процессов вам известны?
14. Какие состояния процессов вам известны? Изобразите диаграмму перехода процессов из одних состояний в другие.
15. Какие дисциплины очередей используются в ОС, каковы их достоинства и недостатки?
16. Что такое управление памятью?
17. Охарактеризуйте режимы MFT и MVT.
18. Что такое страничная память? Приведите примеры организации страничной памяти.
19. Какие типы пользовательских интерфейсов вам известны?
20. Какие элементы диалогового окна используются для управления приложениями?
21. Что такое виджеты? Перечислите известные типы виджетов.
22. Укажите основные элементы интерфейсов, входящих в экран, примеры которых приведены на рис. 1.30, 2.17, 3.14.

Глава 2

ОПЕРАЦИОННЫЕ СИСТЕМЫ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ — ОДНОПОЛЬЗОВАТЕЛЬСКИЕ, ОДНОЗАДАЧНЫЕ И МНОГОЗАДАЧНЫЕ

Как это отмечалось выше, по своим функциональным свойствам ОС могут быть разделены по меньшей мере на три группы:

- *однопользовательские однозадачные;*
- *однопользовательские многозадачные;*
- *многопользовательские многозадачные.*

Несмотря на то, что в исторической последовательности первыми на сцену вышли (в 60—70-е годы XX в.) именно *многопользовательские многозадачные* ОС (OS 360/370, RSX и пр.), наибольшее распространение (вместе с ПК) получили *однозадачные* ОС (MS DOS и ее аналоги).

В настоящей главе предполагается рассмотреть: ОС для ПК — MS DOS, графическую оболочку Windows 3.1.x, а также ОС Windows 95/98/ME, NT/2000. Надо отметить, что начиная с Windows 3.11 for Workgroups данные ОС и оболочки начинают приобретать *многопользовательский многозадачный* характер.

Тем не менее все они рассматриваются в настоящей главе в связи с тем, что в подавляющем большинстве случаев данные ОС, несмотря на свои потенциальные возможности, эксплуатируются в *однопользовательском* (пусть и *многозадачном*) режиме. Что же касается *многопользовательских многозадачных* ОС, то они будут описаны в следующей, 3-й главе.

2.1. Операционная система MS DOS

История операционной системы MS DOS



«Биография» операционной системы MS DOS, получившей широчайшее распространение во всем мире и использовавшейся в количестве, по разным оценкам, от 100

до 150 млн экземпляров, начинается со скромной системы 86-DOS, написанной в середине 80-х годов Т. Петерсоном для компании Seattle Computer Products. При разработке 86-DOS были учтены требования совместимости с весьма популярной в то время системой CP/M-80, предназначенной для восьмиразрядных микрокомпьютеров на базе процессоров Intel 8080 и Zilog Z-80. В результате и в нынешних вариантах MS DOS можно найти немало структур данных и программных средств, характерных для CP/M-80 [15].

В июле 1981 г. фирма Microsoft приобрела права на систему 86-DOS, существенно переработала ее и выпустила на рынок под названием MS DOS (Microsoft Disk Operating System). Когда осенью 1981 г. появились первые персональные компьютеры фирмы IBM, система MS DOS 1.0 и ее аналог фирмы IBM PC DOS 1.0 быстро стали основными системами для этих машин. В то же время непрерывное развитие аппаратных средств компьютеров и накопление опыта работы с ними привели к необходимости столь же непрерывного совершенствования исходных систем MS DOS и PC DOS. В дальнейшем они развивались параллельно, и их новые версии соответствовали друг другу.

Первое серьезное усовершенствование MS DOS (версия 2.0) было выполнено в 1983 г. Фактически была выпущена новая операционная система, хотя разработчикам удалось обеспечить полную совместимость с MS DOS 1.0. В систему MS DOS 2.0 были включены следующие новшества:

- поддержка дискет с повышенной плотностью записи и, главное, появившихся к этому времени жестких дисков;
- иерархическая структура каталогов (пришедшая из системы Unix) вместе с группой команд ее поддержки (`cd`, `md`, `rd` и др.);
- перенаправление ввода-вывода, конвейеры и фильтры (средства, тоже характерные для системы Unix);
- утилита `print`, обеспечивающая вывод на печать в фоновом режиме с возможностью одновременного выполнения любой программы;
- атрибуты файлов и их системная поддержка (в частности, команда `attrib`);
- метка тома и соответственно команды `label` и `vol`;
- устанавливаемые драйверы внешних устройств;
- драйвер `ansi.sys` для расширения возможностей экрана и клавиатуры;

- файл конфигурирования `config.sys`;
- поддержка блоков окружения и соответственно команда `set`;
- динамическое выделение и освобождение памяти;
- поддержка национальных форматов;
- расширение возможностей командных файлов (команды `echo`, `for`, `goto` и др.);
- большая группа новых команд, утилит и драйверов устройств (`backup`, `restore`, `exit`, `find`, `keyb`, `path`, `prompt`, `set`, `vdisk.sys` и др.).

Система MS DOS 3.0 появилась в августе 1984 г., одновременно с выпуском компьютеров IBM PC/AT на базе процессоров 80286. Начиная с этой версии в MS DOS входит поддержка расширенной памяти, жестких дисков увеличенного объема, разделяемых файлов (команда `share`).

Начиная с версии 3.1, выпущенной в ноябре 1984 г., в MS DOS включается поддержка сетевых структур.

В версиях MS DOS 3.2 и особенно 3.3 получили дальнейшее развитие возможности установки национальных форматов, введена поддержка дискет диаметром 3,5 дюйма и жестких дисков с емкостью более 32 Мбайт за счет создания на них нескольких разделов по 32 Мбайт (или менее) каждый, включен ряд новых команд и утилит (`append`, `call`, `chcp`, `fastopen`, `nlsfunc`, `replace`, `xcopy`), а также драйверов устройств (`display.sys`, `driver.sys`).

В 1988 г. появилась версия MS DOS 4.0, для которой фирма Microsoft разработала собственную оболочку `shell` (в версии 4.01 был разработан русифицированный вариант оболочки). Кроме этого, в версию 4.01 включена поддержка разделов на жестких дисках, превышающих 32 Мбайт, средства эмуляции дополнительной памяти, а также ряд новых команд (`append`, `mem`, `truname`).

В MS DOS версии 5.0 существенно улучшена поддержка расширенной и дополнительной памяти, усовершенствована оболочка `shell`, включен улучшенный интерпретатор `qbasic` (вместо утилит `basic` и `basica` предыдущих версий), добавлен ряд новых команд, утилит и драйверов (`doskey`, `edit`, `fc`, `help`, `mirror`, `setver`, `undelete`, `unformat`, `himem.sys`, `ramdrive.sys`, `smartdrv.sys`). Пожалуй, наиболее привлекательной чертой MS DOS 5.0 явилась возможность организации на компьютерах с расширенной памятью специальных областей — области старшей памяти (НМА) и блоков верхней памяти (UMB), куда мож-

но загружать устанавливаемые драйверы, резидентные программы и большую часть самой DOS. Это позволило существенно увеличить объем памяти, отводимой прикладным программам (до 600—610 Кбайт).

Операционная система MS DOS 6.0, выпущенная в 1993 г., вобрала в себя все лучшие качества предыдущих версий. Из MS DOS 6.0 удалены некоторые устаревшие средства (*assign*, *backup*, *comp*, *edlin*, *grftabl*, *join*, *mirror*, *recover*, *truname*). В то же время в систему был включен ряд полноэкранных инструментальных утилит, охватывающих практически весь диапазон потребностей пользователей персональных компьютеров. Утилиты имели развитый интерфейс пользователя, управлялись как от клавиатуры, так и мышью, включали контекстные справочники и элементы обучающих систем. В MS DOS 6.0 входили следующие утилиты:

- оболочка MS DOS *shell* — многофункциональная программа, существенно упрощающая работу пользователя с файлами, каталогами и программами и предоставляющая ему ряд дополнительных возможностей, отсутствующих в самой DOS, например объединение программ в программные группы, защита их паролем, организация многозадачного режима с удобным переключением между задачами и другие;
- утилита резервного копирования *msbackup*, осуществляющая получение резервных (архивных) копий файлов жесткого диска на архивных дискетах. Утилита обеспечивает все основные режимы резервного копирования (полное, инкрементное и разностное) и отличается высокой эффективностью;
- утилита *defrag*, служащая для оптимизации файловой структуры на диске путем дефрагментации файлов и диска в целом;
- антивирусная программа-утилита *msav*, позволяющая вылечить диск, инфицированный вирусами;
- утилита *msd* для получения технической информации о вычислительной системе;
- системный отладчик *debug*, позволяющий отлаживать и изучать работу выполнимых программ, а также выполнять операции с памятью и портами компьютера;
- текстовый редактор MS DOS *edit*, позволяющий создавать и редактировать текстовые файлы;

- интерактивный полноэкранный интерпретатор с языка Бейсик `qbasic`;
- утилита `memmaker`, позволяющая организовать оптимальное использование наличной памяти;
- утилита сжатия диска `dblspace`, осуществляющая сжатие (компрессию) файлов в процессе их записи на диск и автоматическое разворачивание при загрузке в память, что позволяет существенно увеличить эффективную емкость диска.

Состав команд MS DOS 6.0 в целом совпадает с предыдущими версиями, этим обеспечена совместимость версий DOS на уровне интерфейса пользователя. С другой стороны, многие команды DOS приобрели дополнительные свойства; добавлен ряд новых команд (`choice`, `fasthelp`, `loadfix`, `numlock`, `power`, `vsave` и др.). Существенно развиты средства межмашинной связи (драйвер `int-erlnk.exe` и команды `interlnk` и `intersvr`).

Важнейшим усовершенствованием, введенным в версию MS DOS 6.0, является возможность задания в процессе начальной загрузки альтернативных конфигураций системы (методика использования расширенной и дополнительной памяти, состав загружаемых драйверов устройств, наличие и характеристики электронных дисков и пр.). Альтернативное конфигурирование осуществляется с помощью специальных директив файла `config.sys`. Данная версия операционной системы MS DOS 6.0 является весьма совершенным программным продуктом, обеспечивающим эффективное использование персональных компьютеров всех моделей — от исходных IBM PC и PC/XT до более поздних PC/AT и PS/2 на базе процессоров 80386, 80486 и Pentium, оборудованных расширенной памятью, магнитными и лазерными дисками большого объема и средствами межмашинной связи.

Совместимость операционных систем. Обычно системное программное обеспечение DOS подстраивается к конкретной машине. При этом оно конструируется так, чтобы могло подойти для любой машины, совместимой с данной (например, для операционных систем PC DOS или MS DOS версий COMPAQ или Cordata). Единственная область, где программное обеспечение разных операционных систем дифференцировано, это файл `io.sys`. Он непосредственно связан с физическим устройством электронного оборудования и организуется независимо каждой фирмой-изготовителем. Однако электронное оборудование разных систем сходно по своему строению, и это обеспечивает совместимость `io.sys` по основным параметрам.

Благодаря такой совместимости пользователь может без затруднений сменить операционную систему на своей вычислительной машине. Однако при переходе в другую систему следует помнить, что ее системные файлы, как правило, отличаются по размеру от системных файлов системы, работавшей ранее. Если системные файлы новой системы больше системных файлов предыдущей (не укладываются в отведенное предыдущей системой место), то переход в новую операционную систему может вызвать затруднения.

Некоторые основные понятия, связанные с функционированием MS DOS

Устройства ПЭВМ зарегистрированы в ОС под зарезервированными именами, типовые значения и содержание которых следующие (см. также табл. 2.1):

- накопители на гибких магнитных дисках (НГМД) обозначаются латинскими символами `a:`, `b:`, емкость ГМД от 180 Кбайт до 2,8 Мбайт;
- накопители на жестких магнитных дисках (НЖМД) обозначаются символами `c:`, `d:`, емкость от 100 Мбайт до 100 Гбайт;
- `lpt` — принтер;
- `con` — клавиатура ЭВМ (консоль);
- `comx` — последовательный порт (разъем, предназначенный для подключения устройства, коммутируемого по предусмотренным стандартам, например аналогичной ПЭВМ).

С точки зрения пользователя эти устройства (табл. 2.1) ничем не отличаются от обычных файлов (с ними можно производить те же операции, что и с обычными файлами).

Набор данных (файл) — именованная совокупность данных, размещаемых на НГМД или НЖМД. Наименование файла включает *имя и расширение*, строящиеся по принципам, аналогичным принятым для других ОС.

В полном имени файла разрешается использовать только следующие символы:

`A — Z 0 — 9 $ & # ` ~ () - % ! _ ^`

В полном имени файла запрещается использовать все остальные символы, а также пробел.

Таблица 2.1. Стандартные устройства, зарегистрированные в MS DOS

Имя	Назначение файла
AUX	Асинхронный интерфейс
CLOCK\$	«Часы»
COM1	Порт последовательного ввода-вывода
COM2	
COM3	
COM4	
LPT1	Порт параллельного ввода-вывода
LPT2	
LPT3	
CON	Консоль (клавиатура)
NUL	«Нулевое устройство»
PRN	Принтер (аналог LPT1)

Примеры допустимых имен файлов:

Format.com, Read.me, MyFile.txt, 28-03-99.doc, 123.45.

Примеры недопустимых имен файлов:

123456789.txt, aa?.doc, 35*.? It.F.doc.txt

Наиболее типичные значения расширений отражают следующие типы файлов (см. также табл. 1.2):

- .com, .exe, .bin — исполняемые программные модули;
- .bat — пакетный (командный) файл;
- .txt — текстовый файл;
- .arc — архивный файл.

Каталог (директория) — именованная логическая область на диске, содержащая группу файлов, которая в свою очередь включает подкаталоги (субдиректории). Совокупность каталогов образует на диске иерархическое дерево каталогов, при этом единственная директория, не входящая в другие, называется корневой. При инициализации диска на нем создается только корневой каталог.

В отличие от корневого каталога, остальные каталоги (подкаталоги) создаются с помощью команд MS DOS. Основная цель такой структуры каталогов — организация эффективного хранения большого количества файлов на диске. Каждый каталог (кроме корневого) «имеет» «родителя», т. е. другой каталог, к которому «привязан» данный каталог (термин «привязан» иногда заменяется термином «зарегистрирован»). MS DOS рассматривает каждый каталог (кроме корневого) как файл (рис. 2.1).

На рис. 2.1 приводится пример дерева каталогов, отображаемого программной оболочкой Norton Commander (в режиме эмуляции DOS, см. также рис. 4.33).

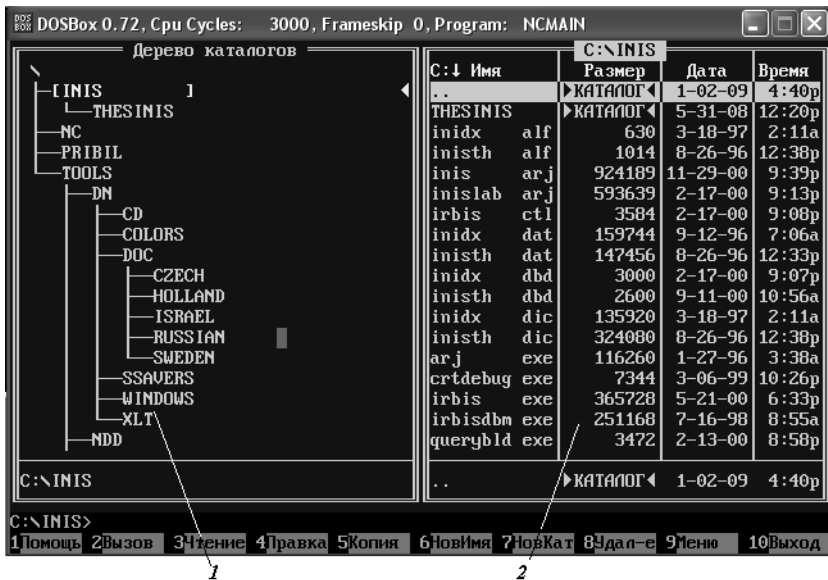


Рис. 2.1. Экран Norton Commander.

1 — дерево каталогов (левая панель); 2 — содержание каталога INIS (правая панель)

Спецификация файла есть совокупность обозначений, обеспечивающих поиск данных на диске, и имеет следующий вид:

устройство\путь\имя.расширения

Здесь путь — список субдиректорий, входящих друг в друга, в младшей из которых, собственно, содержится файл.

Примеры:

a:\command.com — программный файл, содержащийся в корневой директории устройства a:;

b:\sapr\acad.exe — программный файл, входящий в директорию sapr диска b:;

c:\sapr\graph\graphed.com — программный файл, входящий в подкаталог graph 2-го порядка. Здесь путь — \sapr\graph.

Так называемые символы подстановки позволяют «фильтровать» файлы, используя их имена. К ним относятся символы ? и *. Эти символы можно использовать в любом месте собственно имени файла (имени и расширении).

Символ ? означает, что команда (при фильтрации файлов) принимает любой символ в имени или расширении файла, в позиции которого находится символ ?.

Символ * означает, что команда (при фильтрации файлов) принимает цепочку символов в имени или расширении файла, начиная с позиции, где находится символ *.

Символы ? и * действуют независимо друг от друга применительно к имени или расширению.

Примеры.

Выполнить операцию над следующими группами файлов:

. — все файлы без исключения;

*.txt — файлы с любым именем, но с расширением .txt;

gg*.* — файлы, имена которых начинаются с цепочки символов gg и имеют любое расширение;

YE??0198.* — файлы, имена которых начинаются с цепочки символов YE, два следующих символа не имеют значения, следующие четыре символа должны быть 0198, расширение — произвольное.

Текущий дисковод — устройство, адрес которого подразумевается по умолчанию, при отсутствии его явного указания пользователем. Обычно текущее (или активное) устройство указано в подсказке, выводимой системой на экран, при готовности приема команд пользователя, которая имеет обычно вид:

a:>, b:> и т. д.

Текущий дисковод устанавливается командой a:, b: и пр.

Текущая директория, каталог (субдиректория) или текущий путь — совокупность вложенных каталогов, подразумеваемая по умолчанию при поиске файла, если пользователь явно не задаст

ее в спецификации. Обычно описывается в подсказке типа A:\ACAD>. Устанавливается командой `cd`.

Управление вводом-выводом. Ввод и вывод — это процессы, осуществляющие пересылку входных и выходных данных. MS DOS предусматривает достаточно сложное программное обеспечение для управления этими процессами по желанию пользователя. Управление данными осуществляется с помощью процедур, называемых направленным вводом и выводом, фильтры и коммуникации.

Стандартные устройства ввода-вывода. Для ввода информации в большинстве случаев используют клавиатуру. В результате выполнения большинства операций полученные данные выводятся на экран дисплея. Поэтому клавиатура считается стандартным устройством ввода, а экран — стандартным устройством вывода.

Запуск вычислительного процесса в MS DOS осуществляется путем ввода (полностью или частично) спецификаций программного (типа `.exe`, `.com`, `.bin`) или пакетного (`.bat`) файла, расположенного в текущем каталоге текущего устройства.

Основные команды для работы с каталогами: `dir`, `mkdir`, `rmdir`, `chdir`.

Основные команды для работы с файлами: `type`, `delete`, `copy`, `rename`.

Основные команды для работы с дисками: `format`, `diskcopy`, `vol`, `label`, `chkdsk`, `sys`.

Основные команды конфигурирования системы и управления устройствами: `cls`, `date`, `path`, `prompt`, `time`, `ver`.

Перечислим наиболее употребительные команды, не выделяя специально внутренние и внешние (подробный алфавитный перечень приведен в табл. 2.2).

`format` — подготовка диска к работе (форматирование или инициализация). Команда внешняя, поэтому подразумевается наличие файла `format.com`. Команда предусматривает ключи:

`v` — форматирование с записью на диск метки (имени) тома;

`s` — перенос операционной системы с резидентного на форматруемый диск.

`mkdir (md)` — создание каталога (подкаталога). Аргументом является имя создаваемого каталога.

`copy` — копирование файлов, их групп и обмен между устройствами ЭВМ. Формат команды:

`copy исходная_спецификация результирующая_спецификация`

Таблица 2.2. Алфавитный перечень основных команд MS DOS
(символом * в первой колонке отмечены внутренние команды)

Команда	Функция
Ansy.sys	Установка драйвера консоли
Assign	Переназначение дисковых устройств
Attrib	Установка атрибута файла
Batch*	Пакетные командные файлы (*.bat)
Backup	Создание резервных копий для файлов
Break*	Прерывание программы
Buffers*	Создание буферов в ОП
Chdir (cd)*	Переход в новый каталог
Chkdsk	Проверка дисков
Cls*	Очистка экрана
Command*	Второй командный процессор
Comp	Сравнение дисковых файлов
Copy*	Копирование файла
Country	Установка формата даты и времени
Ctty	Переназначение консоли
Date*	Установка даты
Debug	Отладчик программ
Device*	Установка новых драйверов устройств
Dir*	Просмотр каталогов
Diskcomp	Сравнение дисков
Diskcopy	Дублирование дискет
Driver.sys	Установка драйвера блочно-ориентированных устройств
Erase (del)*	Удаление файлов
Fcbs*	Блоки управления файлами
Fdisk	Разбиение жесткого диска на разделы
Files*	Установка числа одновременно открытых файлов
Find	Поиск данных
Format	Форматирование диска
Graftable	Загрузка дополнительных символов для графического режима
Graphics	Распечатка графических изображений

Окончание табл. 2.2

Команда	Функция
Join	Логическое объединение каталога на одном диске с другим диском в один каталог
Keybxx	Загрузка нерезидентных драйверов клавиатуры
Label	Создание и замена метки диска
Lastdrive*	Установка максимального числа доступных дисководов
Link	Загрузчик (редактор) связей
Mkdir (md)*	Создание каталога
Mode	Изменение режимов работы выходных устройств
More	Постраничный вывод файлов на экран
Path*	Указание пути поиска
Print	Вывод на печать данных
Prompt*	Изменение формата приглашения DOS
Rename (ren)*	Переименование файлов
Replace	Селективная замена и копирование файлов
Restore	Восстановление файлов, резервированных по команде BACKUP
Rmdir (rd)*	Удаление пустого каталога
Select	Установка MS DOS на новый диск с заданным типом клавиатуры, форматом даты и времени
Set*	Установка переменной окружения
Shell*	Применение дополнительного командного процессора
Sort	Сортировка данных
Subst	Создание виртуальных дисков
Sys	Копирование MS DOS
Time*	Установка времени
Tree	Вывод дерева каталогов
Type*	Вывод на дисплей содержимого файла
Vdisk.sys	Установка драйвера виртуального диска
Ver*	Вывод версии MS DOS
Verify*	Проверка записи на диск
Vol*	Вывод метки диска
Xcopy	Выборочное копирование групп файлов и каталогов

Примеры:

copy a:acad.exe b:\sapr — копирование с диска на диск в заданный каталог;

copy *.* b: — копирование всех файлов из текущего каталога текущего устройства на диск b:;

copy con lpt — копирование текста, вводимого с клавиатуры, на печать;

copy *.pas lpt — распечатка всех ПАСКАЛЬ-программ;

copy com1 a: — прием данных через последовательный порт внешней связи и запись на дисковод a:;

erase (del) — удаление файлов или их групп;

del a:*.for — удаление с диска всех ФОРТРАН-программ;

rmdir (rd) — удаление каталога, предварительно вычищенного командой del;

установка текущего устройства — ввести имя устройства, например a:;

chdir (cd) — изменить (установить) текущую директорию;

dir — вывести на экран оглавление каталога;

ren — переименование файла;

type — вывод файла на экран монитора.

В MS DOS также присутствует категория командного или пакетного файла, состоящего из командных строк MS DOS, наименований пользовательских программ и командных слов (операторов) типа:

if — проверка условия;

goto — передача управления указанной инструкции внутри файла;

for — многократное применение одной и той же команды;

pause — остановка процесса до нажатия пользователем какой-либо клавиши.

Основные составные части MS DOS

MS DOS состоит из следующих компонент:

- базовой системы ввода-вывода (BIOS);
- блока начальной загрузки;
- модуля взаимодействия с BIOS (io.sys для версии 5.0 и выше);
- модуля обработки прерываний (msdos.sys для версии 5.0 и выше);

- командного процессора (`command.com`);
- внешних команд (утилит) MS DOS;
- драйверов устройств;
- файла `config.sys`;
- файла `autoexec.bat`.

Базовая система ввода-вывода (BIOS). Модуль BIOS индивидуален для каждой вычислительной системы и поставляется ее изготовителем. В этом модуле содержатся аппаратно-зависимые драйверы следующих устройств:

- консоль (монитор) с клавиатурой (`con`);
- устройство печати (`prn`);
- последовательный канал связи (`aux`);
- часы/календарь (`clock`);
- дисковое устройство начальной загрузки (блочно-ориентированное устройство).

BIOS находится в постоянном запоминающем устройстве (ПЗУ) и эту систему можно рассматривать, с одной стороны, как составную часть аппаратных средств, а с другой — как один из программных модулей DOS.

Основные функции BIOS:

- автоматическое тестирование основных аппаратных компонентов при включении машины, в том числе оперативной памяти;
- вызов блока начальной загрузки DOS по окончании тестирования. Поскольку DOS состоит из нескольких модулей, загрузка ее в память проходит в два этапа; сначала BIOS загружает с системного диска в оперативную память специальный блок начальной загрузки, а затем уже передает на него управление, а тот, в свою очередь, осуществляет загрузку других модулей DOS;
- обслуживание системных вызовов или прерываний. Системные вызовы вырабатываются программными или аппаратными средствами с целью выполнения различных операций. Для реализации системных вызовов используется механизм прерываний. Суть этого механизма заключается в том, что текущая работа машины может быть приостановлена на короткое время одним из сигналов, который указывает на возникновение ситуации, требующей немедленной обработки. Прерывания можно разделить на три группы: аппаратные, логические и программные. Каждое прерывание имеет свой уникальный номер и с ним может быть свя-

зана определенная подпрограмма, призванная обслуживать возникшую ситуацию.

Таким образом, BIOS является своеобразной программной оболочкой аппаратных средств ПК, предоставляя возможность другим программам, в частности самой DOS, обращаться к аппаратным компонентам через механизм прерываний.

Блок начальной загрузки (БНЗ, Boot Record), или просто загрузчик — это небольшая программа, единственная функция которой заключается в считывании с дисков в оперативную память двух других частей DOS — модуля расширения базовой системы ввода-вывода (`io.sys`) и модуля обработки прерываний (`msdos.sys`). Для упрощения БНЗ они размещаются на системном диске всегда первыми — один за другим в последовательных секторах. При этом оба файла снабжаются атрибутом «скрытый файл» (`hidden`), который делает их «невидимыми» при выводе каталога диска на экран дисплея или на принтер. Длина БНЗ — 512 байт, т. е. он занимает всего один сектор на диске, и уже поэтому содержащаяся в нем программа не может быть чрезмерно сложной.

Если БНЗ не обнаруживает указанных файлов на первых двух позициях каталога диска, то данный диск считается «несистемным», о чем на экран выдается соответствующее сообщение. Пользователь в этом случае может вставить в дисковод «правильный» системный диск и, нажав любую клавишу, подтолкнуть БНЗ к повторению поиска DOS на вновь установленном диске. Здесь следует заметить, что машину можно «перезапустить» двумя способами. Один из них, очевидный, заключается в выключении и повторном включении питания всей машины («холодный перезапуск»). Однако этот «грубый» способ не очень-то полезен для электронных компонентов системного блока, чувствительных к броскам напряжения питания. Другой способ перезапуска («теплый») производится с помощью комбинации клавиш `<Ctrl+Alt+Del>`. В этом случае загрузка системы начинается с BIOS; при этом тестирование памяти и других аппаратных компонентов не производится, а сразу начинается поиск и загрузка в память БНЗ.

Заметим, что БНЗ всегда помещается на диск при его начальной разметке (форматировании) независимо от того, записываются ли на диск также и «системные» файлы с другими модулями DOS. Что касается двух других упомянутых системных файлов, то они появляются на диске лишь при форматировании командой `format` с ключом `/s`. Занести их на уже заполненный

диск (с непустым каталогом файлов) нельзя, так как начальные позиции каталога и начальные сектора диска уже будут заняты другими файлами. Если же диск размечен обычным образом, но еще пуст, то можно поместить на него системные файлы, применив утилиту DOS `sys.com`.

Модуль расширения базовой системы ввода-вывода (`io.sys`). Поскольку BIOS является инвариантной по отношению к операционной системе, устанавливаемой на данной машине, расширение BIOS с помощью дополнительного модуля `io.sys` придает гибкость операционной системе, позволяя управлять с ее помощью таким набором аппаратных средств ПЭВМ, который наиболее точно соответствовал бы замыслу разработчиков операционной системы. Этот модуль относительно легко может модифицироваться разработчиками с учетом нужд конкретной версии DOS.

Появляется возможность включения в BIOS дополнительных подпрограмм, обслуживающих новые внешние устройства (драйверов). Драйверы разрабатываются не только для новых внешних устройств, но и для тех, которые стандартно входят в состав аппаратуры в тех случаях, когда обмен информацией с ними должен происходить иначе, чем принято в стандартной версии DOS. Это придает системе большую гибкость и возможность адаптации ее под любые требования пользователей.

О необходимости подключения новых драйверов внешних устройств, а также об изменении других параметров DOS уведомляется через файл конфигурации `config.sys`. Этот текстовый файл обрабатывается модулем расширения `io.sys`, который осуществляет необходимую подстройку прерываний и других параметров DOS в соответствии с заданными в `config.sys` командами конфигурирования.

Кроме рассмотренных выше функций, на модуль `io.sys` возлагается еще одна задача — завершение загрузки DOS в оперативную память. С этой целью модуль расширения сначала передает управление на загруженный к этому моменту в оперативную память модуль обработки прерываний DOS, а после этого управление возвращается в модуль `io.sys`, который производит загрузку командного процессора с диска в оперативную память и передает ему управление.

Модуль обработки прерываний (`msdos.sys`). В отличие от модуля нижнего уровня `io.sys`, модуль обработки прерываний `msdos.sys` образует верхний уровень системы, с которым взаи-

действует большинство прикладных программ (иногда этот модуль DOS называют основным или ядром системы).

Компонентами данного модуля являются подпрограммы, обеспечивающие работу файловой системы, устройств ввода-вывода (клавиатуры, монитора, принтера, коммуникаций), обслуживание некоторых специальных ситуаций, связанных с завершением программ, их искусственным прерыванием и обработкой ошибок. Некоторые из этих подпрограмм довольно велики по объему.

Деление сервисных функций DOS на два уровня обусловлено соображениями модульности и будущего развития системы. Функции, реализуемые модулем `msdos.sys`, в первую очередь используются командами DOS (обрабатываемыми командным процессором: `dir`, `copy` и др.), но с тем же успехом они могут вызываться прикладными программами.

Командный процессор (интерпретатор команд, `command.com`). В отличие от модулей `io.sys` и `msdos.sys`, файл с командным процессором `command.com` может занимать на системном диске любое место и трактуется как обычная программа.

Основные функции командного процессора:

- прием и разбор команд, полученных с клавиатуры или из командного файла;
- исполнение встроенных (внутренних) команд DOS;
- загрузка и исполнение внешних программ DOS и прикладных программ (находящихся в файлах типа `.com` и `.exe`). Запуск прикладной программы осуществляется так же, как и обращение к внешней команде DOS (утилите), поскольку те и другие реализованы одинаково;
- исполнение файла автозапуска `autoexec.bat`.

Когда в качестве команды DOS командный процессор встречает имя файла, не совпадающее с именами встроенных команд, производится анализ типа этого файла, указанного в каталоге. Файлы типов `.com` и `.exe` считаются загрузочными и обрабатываются соответствующим образом, а файл типа `.bat` трактуется как командный.

Исполняемые файлы типа `.com` не требуют настройки адресов после их загрузки в оперативную память, а программы типа `.exe` при загрузке с диска в оперативную память необходимо настроить по месту размещения, т. е. задать соответствующие адреса сегментов.

Когда в качестве команды DOS поступает имя командного файла, командный процессор начинает последовательно читать

и интерпретировать содержащиеся в нем строки, каждая из которых может содержать одну команду, метку или комментарий. Если в очередной строке стоит команда, осуществляющая вызов некоторой программы, то интерпретация командного файла приостанавливается и начинается исполнение вызванной программы. После ее завершения управление возвращается командному процессору.

Язык команд DOS служит основным средством общения пользователя с дисковой операционной системой. После вызова какой-либо прикладной программы («задачи») пользователь взаимодействует с ней, а не с DOS, вплоть до завершения задачи, после чего вновь вступает в действие командный процессор.

Если в конкретной команде DOS предусмотрено задание неполного списка аргументов и/или флажков, то командный процессор подставляет вместо недостающих параметров определенные значения. Эта операция называется подстановкой по умолчанию.

При загрузке в оперативную память командный процессор разделяется на две области: резидентную, которая располагается вслед за двумя рассмотренными выше модулями DOS, и нерезидентную, которая располагается на старших адресах ОЗУ.

Резидентная часть содержит подпрограммы стандартной обработки определенной группы прерываний, которые разработчик прикладной системы может переопределить по-своему. Здесь же

00500	512		DOS communication area	
00700	1424	IBMBIO		
00C90	5184	IBMDOS		
020E0	35584		System data	
0ABF0	64		System data	
0AC40	16		Data	
0AC60	336	RK	Environment	
0ADC0	32		Data	
0ADF0	6928		Program	
0C910	10880	RK	)**6-ю/0123456789%*5=0...	43 F4
0F3A0	336		Data	
0F500	3536		Program	21
102E0	3360		Program	33 5C 67
11010	400	CONAGENT	Environment	
111B0	14704	CONAGENT	/new	09 10 8E
14B30	384	CONAGENT	Data	
14CC0	416	SHELL	Data	
14E70	5712	SHELL		22 23 24 2E 2F
164D0	2064	SHIFT	Environment	
16CF0	384	MI	Environment	
16E80	16352	MI		00
1AE70	545168		- Free -	

Рис. 2.2. Размещение в оперативной памяти программ после загрузки MS DOS

находится программа подзагрузки нерезидентной части командного процессора в оперативную память. Кроме того, в состав резидентной части входит подпрограмма инициализации, которая интерпретирует файл `autoexec.bat` при запуске машины.

Нерезидентная часть включает программу обработки команд DOS, поступающих с терминала или из командных файлов. В нерезидентную часть входит загрузчик внешних программ, задачей которого является чтение очередной программы с диска в оперативную память, настройка адресов и переход на ее исполнение.

После загрузки и инициализации командного процессора оперативная память разделяется на три области (см. рис. 2.2):

- область DOS — от младших адресов, объемом около 60 Кбайт;
- область прикладных программ максимальным объемом около 580 Кбайт;
- системная область — на старших адресах ОЗУ объемом 384 Кбайт.

Утилиты DOS — «внешние» команды (программы), входящие в стандартный комплект DOS в виде отдельных загрузочных файлов и выполняющие сервисные функции.

В DOS входит более десятка утилит, предназначенных для разных целей (см. табл. 2.2).

Достоинством DOS и других операционных систем этого типа является то, что любая программа может играть роль сервисной наравне со стандартными утилитами, поскольку ее запуск не отличается от вызова утилит DOS.

Указанное свойство делает DOS открытой для расширения. Неизменное ядро системы составляет лишь BIOS и основные модули `io.sys`, `msdos.sys`, `command.com`. Общий объем оперативной памяти, занимаемой этими модулями, составляет от 40 до 60 Кбайт в зависимости от конкретной версии системы.

Драйверы устройств — это резидентные программы, которые дополняют систему ввода-вывода DOS и обеспечивают обслуживание новых или нестандартное использование имеющихся устройств. Например, с помощью драйверов возможна работа с «электронным диском», т. е. с частью памяти компьютера, с которой можно работать так же, как с диском. Драйверы загружаются в память компьютера при загрузке операционной системы, их имена указываются в файле `config.sys`. Такая схема облегчает добавление новых устройств, позволяя делать это, не затрагивая системные файлы DOS.

Файл конфигурации системы `config.sys`. Текстовый файл, содержащий информацию о подгружаемых дополнительных драйверах и некоторую другую информацию, касающуюся непосредственно MS DOS и выполняемых в ее среде прикладных программ. MS DOS выполняет этот файл автоматически, сразу после загрузки `command.com`.

Команды конфигурирования могут указывать:

- дополнительные драйверы устройств, которые необходимо подключить, например мыши:

```
device=mouse.sys;
```

- количество файлов, которые могут быть открыты одновременно, например:

```
files=20;
```

- количество буферов для обмена информацией с дисковыми накопителями:

```
buffers=8
```

и т. д.

Таким образом, с помощью команд конфигурирования, заданных в файле `config.sys`, пользователь может задать собственные нестандартные возможности, которые вводятся в DOS на время текущего сеанса (до очередного перезапуска машины).

Файл автозапуска программ при загрузке ОС (`autoexec.bat`). Текстовый файл, содержащий дополнительную настроечную информацию. MS DOS выполняет этот файл автоматически, сразу после выполнения `config.sys`.

Загрузка и инициализация DOS

После включения компьютера автоматически выполняется процедура загрузки DOS. Процессор начинает выполнение программ, записанных в ПЗУ — тестирование памяти и элементов процессора (POST — Power on self-testing — процедура самотестирования после включения питания), после чего пытается прочитать с дискового `a:` самый первый сектор, на котором расположена загрузочная запись (Boot record).

Если при включении компьютера на дисковом `a:` не установлена дискета, происходит обращение к жесткому диску и считывается его первый сектор, в котором располагается главная загрузочная запись (Master boot record), а также таблица разделов жесткого диска. Программа главной загрузочной записи про-

сматривает таблицу разделов, находит активный раздел (которым чаще всего является логический диск `c:`) и считывает самый первый сектор этого раздела, где располагается его загрузочная запись (Boot record). Программа загрузочной записи диска `c:` отыскивает на диске системные файлы `io.sys` и `msdos.sys` и загружает их в оперативную память.

После загрузки в память системных файлов начинается конфигурирование системы. Из корневого каталога загружаемого диска считывается файл `config.sys` и выполняются его директивы: загружаются в память и устанавливаются драйверы, указанные в директивах `device`; выделяются буферы DOS, описанные в директиве `buffers`; расширяется системная таблица файлов в соответствии с директивой `files` и т. д. Таким образом, в процессе конфигурирования определяется объем оперативной памяти, занимаемый резидентной частью DOS и, соответственно, оставшийся объем свободной памяти для загрузки прикладных программ.

После окончания обработки файла `config.sys` с диска считывается и загружается в память командный процессор `command.com`. Командный процессор, получив управление, считывает из корневого каталога загружаемого диска файл `autoexec.bat` (если он существует) и выполняет включенные в него команды. Выполнив последнюю команду этого файла, командный процессор выводит на экран системный запрос и входит в цикл ожидания команд оператора.

При необходимости перезагрузки (например, при «зависании» системы, когда она перестает функционировать и не отвечает на нажатия клавиш клавиатуры) можно выключить машину и тут же включить ее снова. Однако безопаснее для компьютера воспользоваться возможностью «мягкой» перезагрузки, которая реализуется нажатием комбинации клавиш `<Ctrl+Alt+Del>`. Эта команда заставляет ПЗУ BIOS выполнить всю описанную выше процедуру начальной загрузки за исключением тестирования. «Жесткую» перезагрузку можно выполнить, нажав клавишу/кнопку `Reset`.

Файловые системы MS DOS

Общие характеристики FAT. Файловая система FAT (File Allocation Table) была разработана в 1977 г. и первоначально использовалась в операционной системе 86-DOS. Чтобы добиться

переносимости программ и файлов из операционной системы CP/M в 86-DOS, в ней были сохранены ранее принятые ограничения на имена файлов. В дальнейшем 86-DOS была приобретена Microsoft и стала основой для ОС MS DOS 1.0, выпущенной в августе 1981 г. FAT была предназначена для работы с гибкими дисками размером менее 1 Мбайта и вначале не предусматривала поддержки жестких дисков. В настоящее время FAT поддерживает файлы и разделы размеров до 2 Гбайт.

В FAT применяются следующие соглашения для имен файлов:

- имя начинается с буквы или цифры и может содержать любой символ ASCII, за исключением пробела и символов `" / \ [ ] : ; | = , ^ * ?`;
- длина имени не превышает 8 символов, за ним следует точка и необязательное расширение длиной до 3 символов;
- регистр символов в именах файлов не различается и не сохраняется.

Файловая система FAT не может контролировать отдельно каждый сектор, поэтому она объединяет смежные секторы в кластеры (clusters). Таким образом уменьшается общее количество единиц хранения, которыми должна управлять файловая система. Размер кластера в FAT является степенью двойки и определяется размером тома при форматировании диска (табл. 2.3). Кластер представляет собой минимальное пространство, которое может занимать файл, и это приводит к тому, что часть пространства диска расходуется впустую. В состав операционной системы входят различные утилиты (DoubleSpace, DriveSpace), предназначенные для уплотнения данных на диске.

Таблица 2.3. Размеры кластеров НЖМД для FAT

Размер раздела, Мбайт	Размер кластера, Кбайт	Тип FAT
Менее 16	4	FAT12
16—127	2	FAT16
128—255	4	FAT16
256—511	8	FAT16
512—1023	16	FAT16
1—2 Гбайт	32	FAT16

Прежде чем переходить к рассмотрению собственно FAT, следует остановиться на общих принципах организации данных на НЖМД ПК, что охватывает более широкий спектр ФС и ОС, используемых в IBM PC-подобных ПК.

Логические и физические диски. Одно из основных понятий файловой системы — логический диск. В некотором приближении можно считать, что, «с точки зрения» ОС, каждый логический диск — это отдельный накопитель на МД.

В MS DOS каждый логический диск имеет свое уникальное имя, в качестве которого используются буквы английского алфавита от а до z (включительно). Количество логических дисков, таким образом, не более 26. Буквы а: и b: отведены строго под имеющиеся в IBM PC дисководы гибких магнитных дисков (НГМД, FDD). Начиная с буквы с: именуются логические диски (разделы) НЖМД (HDD), затем — дисководы оптических дисков (CD/DVD ROM/RW). В случае если данный компьютер имеет только один НГМД, буква b: пропускается. Только логические диски а: и с: могут быть системными (содержать модули MS DOS)*.

В эпоху «больших» машин ОС (OS 360, RSX и пр.) и соответствующие ФС не предусматривали никаких «логических дисков». Пакет магнитных дисков, размещенный на «вертушке» (дисковом устройстве), предоставлял ОС и пользователям непрерывный участок адресуемого пространства памяти, одинаково доступный для использования во всех его областях. Объем такого участка был ограничен емкостью пакета (7,25, 29, 100, 300, 600 Мбайт).

Первые персональные компьютеры IBM PC были укомплектованы только НГМД и поскольку дискеты позволяют хранить относительно небольшие объемы информации, делить флоппи-диск на части также не имело смысла. Следующие модели компьютера IBM PC XT/AT имели жесткий диск объемом 10 или 20 Мбайт, что тоже не создавало проблем, которые возникли, когда производители НЖМД освоили выпуск дисков объемом более 40 Мбайт, поскольку используемый DOS механизм 16-разрядной адресации секторов не позволял использовать диски объемом, большим, чем 32 Мбайт. В ОС MS DOS 3.30 был предложен выход из создавшегося положения — с помощью утилиты

* Отметим, что эти соглашения относятся к MS DOS и Windows. ОС Linux, например, те же разделы, возможно, регистрирует под именами /dev/sda01, /dev/sda03, /dev/sda04 и т. д.

fdisk разбивать физический диск на логические, каждый из которых не превышал бы 32 Мбайт.

В последующих версиях MS DOS упомянутое ограничение на размер адресуемого дискового пространства диска было преодолено, однако традиция применения разделения физического диска на логические (или разделы, «партиции», partitions) полностью сохранилась и применяется повсеместно независимо от типов устанавливаемых ОС. Существует множество причины, по которым может быть оправдано разделение физического диска, в частности:

- в случае повреждения логического диска уничтожается только та информация, которая находилась именно там;
- реорганизация и выгрузка диска небольшого размера проще и быстрее, чем большого;
- возможно разделение дискового пространства между отдельными пользователями персонального компьютера;
- возможна установка для отдельных логических дисков защиты от записи, что увеличивает защищенность от вирусов и других злонамеренных программ;
- в разных разделах можно разместить несколько различных операционных систем, и в ходе начальной загрузки может быть указан раздел диска, из которого должна производиться загрузка операционной системы.

Общая структура жесткого диска. Первый сектор жесткого диска (рис. 2.3, а) содержит так называемую главную загрузочную запись (Master Boot Record — MBR), которая занимает его начальную часть и является программой, которая во время начальной загрузки с жесткого диска помещается по определенному адресу ОЗУ, после чего ей передается управление.

В конце самого первого сектора жесткого диска располагается таблица разделов (TP) диска (Partition Table), каждый элемент которой содержит следующие данные:

- признак активного раздела (раздел может быть использован для загрузки операционной системы). Диск может содержать одновременно несколько активных разделов, которые могут принадлежать разным ОС;
- номер головки, сектора и цилиндра начального сектора раздела;
- номер головки, сектора и цилиндра последнего сектора раздела;
- относительный номер сектора начала раздела;
- размер раздела в секторах и пр.

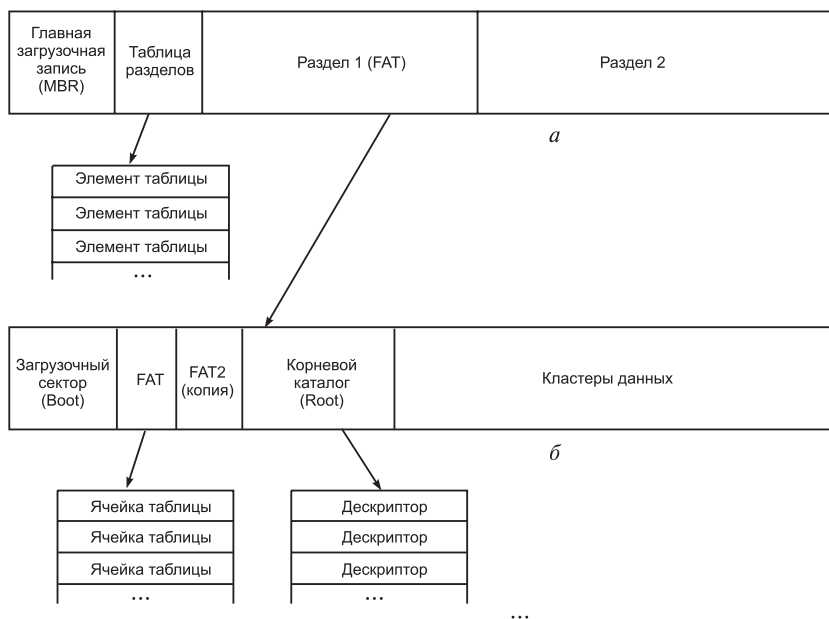


Рис. 2.3. Структура жесткого диска (а); раздел FAT (б)

Загрузка операционной системы с жесткого диска — двухступенчатый процесс. Вначале модули инициализации BIOS считывают главную загрузочную запись (MBR) в память и ей передается управление. Главная загрузочная запись просматривает таблицу разделов и находит активный раздел. Если активных разделов несколько, на консоль выводится сообщение о необходимости выбора пользователем активного раздела для продолжения загрузки.

Такой двухступенчатый метод загрузки операционной системы необходим по той причине, что способ загрузки зависит от самой операционной системы, поэтому каждая операционная система имеет свой собственный загрузчик. Фиксированным является только расположение загрузочной записи — самый первый сектор активного раздела.

Вернемся далее к описанию собственно FAT.

Структура раздела (FAT)

Загрузочная запись BOOT. Первый сектор логического диска (и первый сектор на системной дискете) занимает загрузочная запись (Boot Record). Каждый тип операционной системы имеет свою загрузочную запись.

Кроме программы начальной загрузки операционной системы, в загрузочной записи находятся параметры, описывающие характеристики данного логического диска. Все эти параметры располагаются в самом начале сектора, в его так называемой форматированной области. Формат этой области различается даже для DOS различных версий.

В частности, Boot-сектор для MS DOS версии 4.0 содержит следующие данные:

- команду `jmp` перехода на программу начальной загрузки;
- название фирмы-производителя операционной системы и версию (например, «IBM 4.0»);
- физический номер дисководов;
- серийный номер диска (Volume Serial Number), который создается во время форматирования диска;
- метку диска (Volume Label);
- количество байтов в одном секторе диска;
- количество секторов в одном кластере;
- количество таблиц FAT;
- максимальное количество дескрипторов файлов, содержащихся в корневом каталоге диска;
- количество секторов, занимаемых одной копией FAT;
- количество секторов на дорожке;
- количество магнитных головок и пр.

Поскольку загрузочная запись слишком мала для хранения сложного алгоритма поиска системных файлов, они должны находиться в определенном месте на диске, чтобы загрузочная запись могла их найти. Фиксированное положение системных файлов в начале области данных накладывает жесткое ограничение на размеры корневого каталога и таблицы размещения файлов. Вследствие этого общее число файлов и подкаталогов в корневом каталоге на диске FAT ограничено 512.

Таблица размещения файлов. Сразу после загрузочного сектора на логическом диске находятся секторы, содержащие таблицу размещения файлов FAT (File Allocation Table), от которой и получила свое название данная ФС.

Первоначальная версия FAT, разработанная для DOS 1.00, использовала 12-битовую таблицу размещения файлов (FAT12) и поддерживала разделы объемом до 16 Мбайт (в DOS можно было создать не более двух разделов FAT). Для поддержки жестких дисков размером более 32 Мбайт разрядность FAT была увеличена до 16 бит (FAT16), а размер кластера — до 64 секторов

(32 Кбайт). Так как каждому кластеру может быть присвоен уникальный 16-разрядный номер, то FAT16 поддерживает максимум 2^{16} , или 65 536 кластеров на одном томе.

Первый байт FAT называется «Описатель среды» (Media Descriptor) или байт ID идентификации FAT. Он имеет такое же значение, как и байт-описатель среды, находящийся в Boot-секторе логического диска. Остальная часть FAT состоит из 12- или 16-битовых ячеек, каждая ячейка соответствует одному кластеру диска. Эти ячейки содержат значения, соответствующие следующим характеристикам кластера:

- свободный кластер;
- зарезервированный кластер;
- дефектный кластер;
- последний кластер в списке (конец файла);
- номер следующего кластера в списке (для данного файла).

Очевидно, что здесь мы имеем дело с односписковой структурой, в принципе соответствующей рис. 1.9, *а* (иногда FAT называют файловой системой со связанными списками).

Корневой каталог находится сразу за последней копией FAT и занимает непрерывную область фиксированного размера. Размер корневого каталога задается при форматировании и определяет максимальное количество файлов и каталогов, которые могут быть описаны в корневом каталоге. После корневого каталога на логическом диске находится область файлов и подкаталогов корневого каталога (см. рис. 2.3, *б*). Такую структуру имеют логические диски, расположенные в разделах жестких дисков, а также дискеты.

Область данных разбита на кластеры, причем нумерация кластеров начинается с числа 2. Кластеру с номером 2 соответствуют первые секторы области данных.

Корневой каталог (как и любой другой) состоит из 32-байтовых элементов (дескрипторов), описывающих файлы и вложенные каталоги, которые содержат следующую информацию:

- имя файла/каталога, выравненное влево и дополненное пробелами;
- расширение имени файла, выравненное влево и дополненное пробелами;
- байт атрибутов файла.
- дату и время создания файла (или его последней модификации).
- номер первого кластера, распределенного файлу.
- размер файла в байтах.

Байт атрибутов является принадлежностью каждого файла, и биты этого байта имеют значения, указанные в табл. 2.4.

Таблица 2.4. Кодирование атрибутов файла

Значение	Интерпретация
0	Файл предназначен только для чтения
1	Скрытый файл, этот файл не будет появляться в списке файлов, выдаваемом командой операционной системы DIR
2	Системный файл. Этот бит обычно установлен в файлах, являющихся составной частью операционной системы
3	Данный дескриптор описывает метку диска. Для этого дескриптора поля имени файла и расширения имени файла должны рассматриваться как одно поле длиной 11 байтов (метка диска)
4	Дескриптор описывает файл, являющийся подкаталогом данного каталога
5	Флаг архивации. Если этот бит установлен в 1, то это означает, что данный файл не был выгружен утилитой архивации (например, программой backup)
6—7	Зарезервированы

В любом каталоге, кроме корневого, два первых дескриптора содержат в поле имени следующие строки:

- «.» (точка) — этот дескриптор указывает на содержащий его каталог;
- «..» (две точки) — дескриптор указывает на каталог более высокого уровня.

Таким образом, в древовидной структуре каталогов имеются ссылки как в прямом, так и в обратном направлении, которые применяются для проверки целостности файловой системы.

При удалении файла первый байт его имени заменяется на байт E5h (символ 'x'). Все кластеры, распределенные файлу, отмечаются в FAT как свободные. Сразу после удаления файл еще можно восстановить, так как в дескрипторе сохранились все поля, кроме первого байта имени файла. Но если на диск записать новые файлы, то содержимое кластеров удаленного файла будет изменено и восстановление станет невозможным.

Файловая система FAT всегда заполняет свободное место на диске последовательно от начала к концу. При создании нового файла или увеличении уже существующего она ищет самый первый свободный кластер в таблице размещения файлов. Если в процессе работы одни файлы были удалены, а другие изменились в размере, то появляющиеся в результате пустые кластеры будут рассеяны по диску. Если кластеры, содержащие данные файла, расположены не подряд, то файл оказывается фрагментированным. Сильно фрагментированные файлы значительно снижают эффективность работы, так как магнитные головки при поиске очередной записи файла должны будут интенсивно перемещаться от одной области диска к другой. В состав операционных систем, поддерживающих FAT, обычно входят специальные утилиты дефрагментации диска, предназначенные для повышения производительности файловых операций.

Еще один недостаток FAT заключается в том, что ее производительность существенно зависит от количества файлов, хранящихся в одном каталоге. При большом количестве файлов (около тысячи) выполнение операции считывания списка файлов в каталоге может занять несколько минут. Это обусловлено тем, что в FAT каталог имеет линейную неупорядоченную структуру, и имена файлов в каталогах идут в хронологическом порядке. В результате, чем больше в каталоге записей, тем медленнее работают программы, так как при поиске файла требуется просмотреть последовательно все записи в каталоге.

Поскольку FAT проектировалась для однопользовательской операционной системы DOS, она не предусматривает хранения такой информации, как сведения о владельце или полномочия доступа к файлу/каталогу.

FAT является наиболее распространенной файловой системой и ее в той или иной степени поддерживает большинство современных ОС. Благодаря своей универсальности FAT может применяться на томах, с которыми работают разные операционные системы.

Хотя нет никаких препятствий использовать при форматировании дисков любую другую файловую систему, большинство ОС для совместимости используют FAT. Отчасти это можно объяснить тем, что простая структура FAT требует меньше места для хранения служебных данных, чем остальные системы. Преимущества других файловых систем становятся заметны только при использовании их на носителях объемом более 100 Мбайт.

Надо отметить, что FAT — простая файловая система, не предотвращающая порчи файлов при аварийном завершении работы компьютера. В состав операционных систем, поддерживающих FAT, обычно входят утилиты, проверяющие структуру и корректирующие несоответствия в файловой системе.

Управление оперативной памятью

Память состоит из отдельных элементов, каждый из которых предназначен для хранения минимальной единицы информации — *одного байта*. Каждому элементу соответствует уникальный числовой адрес. Первому элементу присвоен адрес 0, второму — 1 и т. д., включая последний элемент, адрес которого определяется общим количеством элементов памяти минус единица. Обычно адрес задается шестнадцатеричным числом (в тексте шестнадцатеричные числа помечаются заглавной H, например 10H).

Сегменты. Процессор компьютера делит память на блоки, называемые сегментами. Каждый сегмент занимает 64 Кбайт и каждому сегменту соответствует уникальный числовой адрес. Процессор имеет четыре регистра сегмента. Регистр — это участок сверхоперативной памяти процессора, предназначенной для хранения информации. Регистры сегмента предназначены для хранения адресов отдельных сегментов. Они называются CS (сегмент кода), DS (сегмент данных), SS (сегмент стека) и ES (запасной сегмент).

Кроме указанных, процессор имеет еще 9 регистров, а именно — регистры IP (указатель команды) и SP (указатель стека). Регистры CS и IP в паре составляют длинный адрес команды, которая будет выполняться следующей. Регистры SS и SP в паре составляют длинный адрес стека.

Доступ к памяти. Доступ к ячейкам памяти осуществляется посредством соединения содержимого регистра сегмента с содержимым того или другого регистра. Таким образом определяется адрес требуемого участка памяти. Например, адрес следующей команды определяется содержимым регистров CS и IP (записывается «CS:IP»). После выполнения команды и ее удаления из памяти содержимое IP изменяется так, чтобы в регистрах CS, IP находился адрес команды, которая будет выполнена после данной.

Способ объединения регистров для определения адреса ячейки памяти не накладывает ограничений на количество дос-

тупной памяти. Верхнее ограничение зависит от физического строения памяти (т. е. от общего количества ячеек). Первые версии MS DOS разрабатывались для процессора Intel 8088 CPU. Каждый регистр этого процессора рассчитан на хранение 16-битового числа. То есть CPU 8088 комбинирует содержимое сегментного регистра (скажем, CS) с содержимым другого регистра (скажем, IP), получая 20-битовый адрес памяти, что ограничивает доступную память до 2^{20} байтов или 1 Мб.

Доступ к памяти организуется соединением содержимого одного из регистров сегмента с содержимым одного из оставшихся регистров. Значение сегментного регистра называется адресом сегмента. Значение остальных регистров в этом случае называется относительным адресом ячейки памяти (от начала сегмента) или ее коротким адресом. Таким образом, адрес байта вычисляется посредством умножения адреса сегмента на 16 и к полученному значению добавляется короткий адрес.

Сегментные регистры. Сегментные регистры используются при идентификации сегмента памяти. Сегментные регистры применяются в комбинации с регистром указателя или индексными регистрами и в этом случае идентифицируют конкретную ячейку памяти.

Всего сегментных регистров четыре. Регистр CS обычно используется при идентификации блока памяти, в котором хранится код программы. Регистр DS — при идентификации участка памяти, в котором находятся данные этой программы. С помощью регистра SS организуется доступ к стеку. (Стек — это временно выделенная область памяти, обеспечивающая интерфейс «MS DOS—прикладная программа».) Регистр ES — дополнительный (или запасной) сегментный регистр. На него возложены разнообразные функции, часть из которых рассматривается ниже.

Регистры стека. Имеются два регистра стека. Они применяются в комбинации с регистром SS и определяют местонахождение стека. Регистр SP называется указателем начала стека и в комбинации с регистром SS идентифицирует первый байт стека. Регистр BP называется указателем базы стека и в комбинации с регистром SS идентифицирует последний байт стека.

Индексные регистры. Индексных регистров тоже два. Регистры SI и DI применяются в комбинации с одним из сегментных регистров и определяют местонахождение конкретной ячей-

ки памяти. Регистр SI обычно комбинируют с регистром DS, регистр DI — с регистром ES.

Регистры общего назначения. К регистрам общего назначения (их четыре) относятся регистры AX, BX, CX и DX. Это многофункциональные регистры.

Регистр указателя команды. Регистр IP обычно применяется в комбинации с регистром CS и определяет адрес следующей команды.

Регистр флагов состояния. В регистре флагов обычно находятся девять флагов состояния процессора (каждый флаг занимает 1 бит). Эти флаги определяют результат конкретных операций, выполняемых под управлением MS DOS.

Регистры памяти. Регистр памяти включает 2 байта данных (или 16 битов). Реально регистры общего назначения однобайтовые. Так, регистр AX включает регистр AH (который составляет старший байт регистра AX) и регистр AL (который составляет младший байт регистра AX). Аналогично регистры BH, BL, CH, CL, DH и DL — однобайтовые.

Драйверы MS DOS

Два важнейших компонента электронного оборудования компьютера — центральный процессор и память. Остальные компоненты (дисководы, клавиатура, дисплеи, принтеры и т. д.) являются внешними по отношению к компьютеру. Эти внешние компоненты электронного оборудования называются *периферийными устройствами* или просто *устройствами*.

Связь машины с периферийным устройством осуществляется в строго определенном порядке. Каждому периферийному устройству в операционной системе соответствует программа, отвечающая за его контакт с компьютером. Эти программы называются *драйверами*.

Одна из основных функций операционной системы — это обеспечение работоспособности драйверов, доступных системным и прикладным программам. Если работающей программе необходим контакт с периферийным устройством, то она сообщает операционной системе, какое из устройств ей необходимо, и MS DOS предоставляет ей соответствующий драйвер.

Устройства посимвольной и поблочной передачи данных. Устройства посимвольной передачи данных осуществляют пересыл-

ку информации по одному символу. К этим устройствам относятся порты последовательных и параллельных адаптеров и дисплеи. В MS DOS каждому из этих устройств соответствует конкретное название (имя).

2.2. Графические программные оболочки Windows 3.x

Операционная оболочка Windows 3.1



Операционная оболочка Windows 3.1 — надстройка над DOS, обеспечивающая более удобный и наглядный интерфейс для пользователей (графический интерфейс), т. е. набор средств для вывода изображений на экран и манипулирования ими, построения меню, окон на экране и т. д., мультипрограммирования (т. е. возможность одновременного выполнения нескольких программ), имеющая расширенные средства для обмена информацией между программами [16].

В течение долгих лет с момента своего появления персональные компьютеры (IBM-совместимые) обходились без специальных «пользовательских оболочек», работая непосредственно под управлением операционной системы (MS DOS, DR DOS, PC DOS, ...). Пользователи, сядившие за такой компьютер, видели после включения на пустом экране только подсказку `C:\>`. Все операции управления компьютером производились путем ввода с клавиатуры некоторых слов-директив. Неудобство такого алфавитно-цифрового интерфейса порождало претензии и к самим компьютерам (возможно, и не совсем обоснованные).

Основная идея, заложенная в основу оболочки Windows, — естественность представления информации. Информация должна представляться в той форме, которая обеспечивает наиболее эффективное усвоение этой информации человеком. Несмотря на простоту (и даже тривиальность) этого принципа, его реализация в интерфейсах прикладных программ персональных ЭВМ по разным причинам оставляла желать лучшего. Да и реализация его в рамках Windows 3.1 тоже не лишена недостатков. Но эта оболочка представляет собой существенный шаг вперед по сравнению с предыдущими интерфейсами пользователя с ЭВМ. Наиболее важными отличительными чертами ее являются следующие.

Существенно упростилась работа с документами вообще. Можно говорить о документо-ориентированной организации ра-

бот. При этом можно расположить пиктограмму часто используемого документа в окне Диспетчера Программ (Program Manager) и в дальнейшем вызывать процесс обработки этого элемента (например, редактирование) просто двойным щелчком по данной пиктограмме. Той же цели служит и возможность автоматического запуска Диспетчера Файлов после загрузки оболочки — перед пользователем сразу же открывается поле выбора файлов документов.

В среде Windows 3.1 можно составлять документы из частей, которые готовятся в различных приложениях, но при этом сложность работы с таким документом не выше, чем если бы он готовился в рамках одного приложения. Так, при вставке в текст, подготовленный в Write, рисунка, созданного в Paintbrush, рисунок рассматривается как объект. Он может сохраняться, загружаться и печататься совместно с документом. Главной особенностью такого связывания рисунка и текста является простота внесения изменений. Например, работая с текстом в редакторе Write, достаточно выполнить двойной щелчок мышью по рисунку, чтобы вызвать графический редактор Paintbrush. Все внесенные после этого в рисунок изменения автоматически отобразятся и во вставке в текстовый документ.

Работа с объектами предусматривает также и операции с пиктограммами. Пиктограммы можно использовать внутри документов для выполнения функций, подобных тем, которые эти пиктограммы выполняют в среде Windows.

В среде Windows 3.1 реализован новый набор шрифтов — так называемые True Type-шрифты. Эти шрифты похожи на PostScript-шрифты, но легко адаптируются после небольшой настройки практически к любому типу принтера, легко поддаются масштабированию, различного рода деформациям, вращению и т. п.

Наконец, в оболочку Windows 3.1 включены две небольшие мультимедиа-программы (Multimedia). Их использование предполагает наличие соответствующей аппаратной поддержки (акустический адаптер, накопитель CD-ROM). С помощью программ Sound Recorder и Media Player можно оформить процесс прохождения программ звуковыми эффектами, работать с цифровым представлением речи и музыки, с картинками, полученными, например, с проигрывателя видеодисков.

Для начинающих в системе предусмотрена обучающая программа, к которой можно обратиться уже на этапе инсталляции.

Окна в Windows

Рассмотрим способы управления окнами и пиктограммами в Windows 3.1, их структуру и назначения (рис. 2.4).

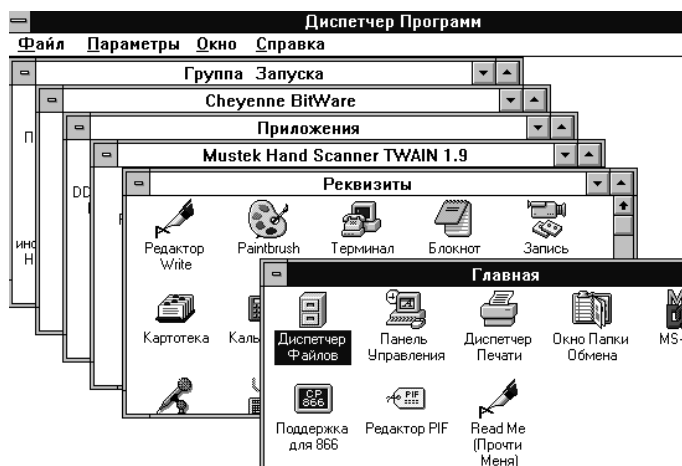


Рис. 2.4. Многооконный интерфейс, типичный для Windows 3.1

Если оболочка Windows запущена и работает, то общение пользователя с ней происходит посредством многооконного интерфейса — через систему окон. Окна или представляющие их пиктограммы расположены на поле экрана, как бумаги на рабочем столе (поле экрана называется Desktop — поверхность стола). Каждое окно в обязательном порядке содержит поле заголовка и рабочее поле (или поле индикации).

Заголовок и рабочее поле окна. Каждое окно представляет собой ограниченную рамкой часть поверхности экрана. Оно может иметь различный размер и находиться в разных местах экрана. Внутри каждого окна что-нибудь показывается (приложением или системой) или что-нибудь делается самим пользователем. Другими словами, окно представляет собой пространство для:

- размещения объектов (текста, рисунков, пиктограмм и других окон);
- выполнения действий (написание текста, рисование, ввод директив, вывод сообщений).

Окно может существовать в полноэкранном представлении, занимая все поле экрана, или в нормальном представлении, занимая только его часть, или в виде пиктограммы.

В среде Windows предусмотрены окна двух типов:

- окна, в которых выполняются приложения (прикладные окна);
- окна, подчиненные другим. В них не выполняются никакие приложения. Они служат для индикации документов или пиктограмм и называются окнами документов или групповыми окнами.

Прикладное окно. Большинство окон, с которыми имеет дело оболочка Windows, — прикладные окна, в которых выполняются конкретные приложения. Распознать прикладное окно легко по двум отличительным признакам (рис. 2.5):

- в заголовке указывается имя приложения, которому принадлежит это окно;
- под заголовком расположена еще одна строка, называемая строкой меню, в которой перечисляется ряд операций, доступных приложению. Как правило, первая из операций — File (Работа с файлами).

Если запустить двойным щелчком приложение Write, представляющее собой редактор текста и входящее в группу Accessories, то открывающееся окно в точности совпадает по виду с приведенным выше на иллюстрации. Но оно, в отличие от уже опробованных приложений, не содержит никакой информации, а только чистое рабочее поле. Предназначение рабочего поля зависит от приложения: в одних приложениях в нем ото-

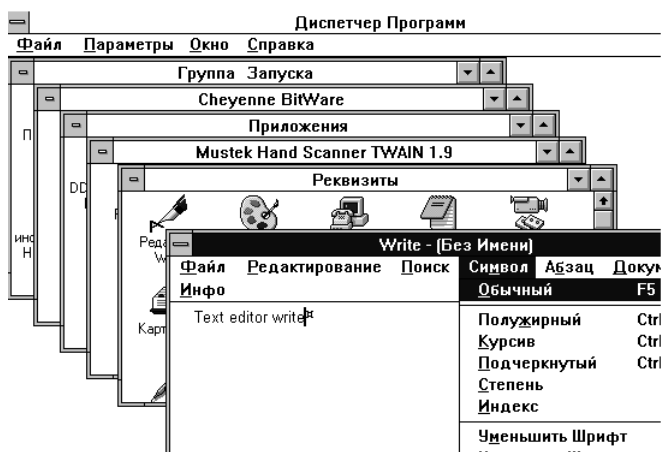


Рис. 2.5. Прикладное окно текстового редактора Write

бражается вводимый с клавиатуры текст, в других — рисуемая посредством мыши картинка, в третьих — результаты расчетов.

Подчиненное окно (групповое окно). Внутри окна Менеджера (диспетчера) программ видно по крайней мере еще одно окно — подчиненное окно Реквизиты (Accessories).

Отличить групповые и подчиненные окна от окон прикладных легко по следующим признакам:

- заголовки окон, подчиненных другим приложениям, как правило, содержат имена документов, которые располагаются в этих окнах. По этой причине такие подчиненные окна называются также окнами документов;
- подчиненные окна не имеют меню;
- подчиненные окна могут перемещаться только внутри окон, которым они подчиняются. Это же касается и операций изменения размера и представления.

Наконец, для групповых окон существует особая клавиатурная комбинация закрытия <Ctrl+F4>.

Если при инсталлировании была заказана полная конфигурация оболочки, то в окне Диспетчера Программ имеется пять подчиненных групповых окон:

- Main — главная группа;
- Games — игры (нет на рис. 2.5);
- StartUp (Группа запуска) — группа автоматически запускаемых при загрузке оболочки программ;
- Accessories (Реквизиты) — группа сервисных программ, инструментарий;
- Applications (Приложения) — группа дополнительных программ (обычно DOS-приложений).

Кроме того, на рис. 2.5 расположены еще две программные группы, созданные пользователями (Mustek Hand Scanner и BitWare).

В любом случае в окне Диспетчера Программ будут находиться либо сами эти групповые окна, либо их пиктограммы. На рис. 2.4, 2.5 приведены все четыре окна в открытом состоянии.

Пиктограммы

Пиктограммы располагаются внутри групповых окон. Они всегда находятся на одном и том же месте в групповом окне независимо от того, запущено ли соответствующее им приложение

или нет. Сопоставляя внешний вид этих пиктограмм с групповыми пиктограммами, можно сделать вывод о том, что окнам разного типа соответствуют пиктограммы разного вида, или о том, что существует несколько разновидностей пиктограмм.

Групповые пиктограммы. Если свернуть некоторое групповое окно (например, Main) до состояния пиктограммы путем щелчка мышью на переключателе пиктограммы (поле справа от заголовка со стрелкой вниз), то на экране появится следующая пиктограмма группы Main.

Групповая пиктограмма — форма представления группового окна. Она располагается всегда в пределах окна-хозяина (если она не видна, то, вероятно, ее загораживает некоторое окно). Подобные пиктограммы существуют еще в пределах окна Диспетчера Файлов.

Пиктограммы всех групповых окон одинаковы и отличаются только подписями.

Программные пиктограммы. Особенностью групповых окон является то, что они содержат в себе не рабочее поле или документ, а набор пиктограмм, посредством которых запускаются приложения. Внешний вид этих пиктограмм, называемых программными пиктограммами, не отличается от тех пиктограмм, которые возникают в нижней части экрана при сворачивании прикладных окон.

Для запуска приложения по программной пиктограмме следует выполнить на ней двойной щелчок мышью.

Программные пиктограммы также существуют только в пределах Диспетчера Программ.

Пиктограммы приложений (прикладные пиктограммы) — результат сворачивания прикладных окон.

Прикладная пиктограмма выглядит точно так же, как и программная пиктограмма, по которой запускалось данное приложение. Но в подписи к прикладной пиктограмме, как правило, присутствует наименование обрабатываемого приложением документа.

Пиктограммы всех трех видов обладают общим свойством: при выполнении двойного щелчка на них открывается соответствующее окно. Но содержимое и назначение открываемого окна в каждом случае свои:

- групповая пиктограмма открывает групповое окно;
- программная пиктограмма открывает окно приложения и запускает соответствующую программу;

- прикладная пиктограмма открывает окно уже работающего приложения (меняет представление прикладного окна).

Составные части окна

Все окна независимо от типа имеют одинаковый состав и структуру. Рассмотрим структуру прикладного окна в нормальном представлении.

Заголовок. Верхняя строка окна (в данном случае — прикладного окна редактора текста) называется заголовком. Она присутствует во всех окнах. Если окно активно, то заголовок этот содержит надпись «белым по черному», т. е. светлым шрифтом на темном фоне. В противном случае надпись в строке заголовка темная на светлом фоне.

На левом конце строки заголовка находится поле (кнопка) вызова системного меню, или системный переключатель. Посредством системного меню можно:

- изменить представление и местоположение окна;
- закрыть окно;
- переключиться на другое окно.

В центре строки заголовка находится имя приложения, выполняемого в данном окне (для прикладных окон), или имя группы (для групповых). Следом за именем приложения в заголовке прикладного окна идет имя обрабатываемого документа (прописными буквами). Например: Write — memo.wri или Paintbrush — winlogo.bmp.

На правом конце строки заголовка располагаются поля размерных переключателей, позволяющих управлять размером (представлением) окна. Левый переключатель иногда называют еще переключателем пиктограммы. Он содержит изображение стрелки вниз . Щелчком мыши на этом поле можно свернуть окно до пиктограммы. Вид правого переключателя  зависит от представления окна — полноразмерное или нормальное.

Строка меню. В окне редактора под строкой заголовка находится еще одна служебная строка — строка меню. Она содержит ряд элементов, начинающийся словом File. Если в окне имеется строка меню, то это прикладное окно.

Строка меню позволяет управлять приложением, выполняемым в окне. При щелчке мышью на элементах этой строки, как

правило, открываются дополнительные меню (подменю), содержащие директивы обслуживания приложения.

Линейки прокрутки. Левую и нижнюю сторону окна окаймляют специальные сервисные средства, называемые линейками прокрутки.

Прокручивать изображение в окне можно, щелкая на полях стрелок, расположенных по краям линеек (окно перемещается по документу в указываемом стрелкой направлении), или перемещая (протаскивая) белый прямоугольник прокрутки (маркер прокрутки) по линейке с помощью мыши.

Вид указателя мыши. В процессе работы с оболочкой (например, при перемещении по экрану) вид курсора мыши может изменяться. Это может быть двойная стрелка различной направленности, черточка, дорожный знак, часы или некоторая пиктограмма.

Курсор мыши указывает на то место, где находится кончик стрелки. В зависимости от точки, на которую указывает курсор мыши, пользователю предоставляется возможность вызвать то или иное действие, например маркировать или переместить некоторый объект.

Маркер перемещается по экрану синхронно с перемещением манипулятора по поверхности стола. Манипулятор можно перемещать в любом направлении — влево, вправо, вверх, вниз и по диагонали. Скорость перемещения курсора по экрану (точнее, скорость реакции его на перемещения манипулятора) может изменяться с помощью специальных средств конфигурирования мыши, которые сосредоточены в приложении Control Panel (Панель Управления).

Для вызова тех или иных действий в большинстве случаев используется левая кнопка манипулятора. Иногда (в особенности в специальных приложениях) используется и правая кнопка. Кроме того, они могут использоваться и совместно. Главные манипуляции, доступные мыши, уже известны:

- фиксация курсора (щелчок) — кратковременное нажатие на левую кнопку мыши;
- двойная фиксация (двойной щелчок) — двухкратное с коротким промежутком кратковременное нажатие левой кнопки мыши;
- перетаскивание (транспортировка, буксировка, Drag & Drop, D'n'D, D&D, «перетащить и бросить») — нажатие левой кнопки мыши и перемещение манипулятора при нажа-

той кнопке. Перетаскивание заканчивается при освобождении кнопки мыши.

Специальные действия. Ориентация оболочки Windows на работу с манипулятором обеспечивает пользователю доступ к достаточно сложным операциям посредством сравнительно простых манипуляций. Простым щелчком можно открыть и закрыть окно или меню. Многие приложения используют пиктографический интерфейс, позволяющий выполнять определенные действия при фиксации курсора на соответствующих пиктограммах.

Двойной щелчок на программной пиктограмме запускает соответствующую программу или открывает ее окно. Тот же двойной щелчок на заголовке окна меняет его представление. Двойная фиксация курсора на имени файла в диалоговом окне загрузки файлов позволяет открыть соответствующий документ.

Меню

Набор средств управления оболочкой Windows не ограничивается пиктограммами, линейками прокрутки и прочими графическими элементами. Эффективным средством управления, доступным практически из любого окна и пиктограммы, является меню. В общем случае меню представляет собой прямоугольное поле, в котором перечислены доступные операции (или доступные для установки опции). Как правило, обозначения элементов меню состоят из одного-двух слов.

Меню практически постоянно находятся в свернутом состоянии и на экране присутствуют только их заголовки (в строке меню прикладного окна). А может не быть и их (если речь идет о системном меню прикладной пиктограммы). Меню, вызванное из строки меню, разворачивается вниз, а вызванное из прикладной или групповой пиктограммы, расположенной у нижнего края окна, — вверх.

Для вызова любой из перечисленных в меню директив достаточно щелкнуть на ней мышью.

В рамках Windows существует две разновидности меню: системное и операционное. Для получения доступа к системному меню (вызова системного меню) следует щелкнуть мышью на кнопке системного меню. Она располагается всегда в левом верхнем углу окна (в строке заголовка) (рис. 2.6). Форма кнопки системного меню напоминает форму продолговатой клавиши

пробела, и этой ассоциацией можно воспользоваться, чтобы запомнить второй способ вызова системного меню (клавиатурный) — <Alt+Space>.

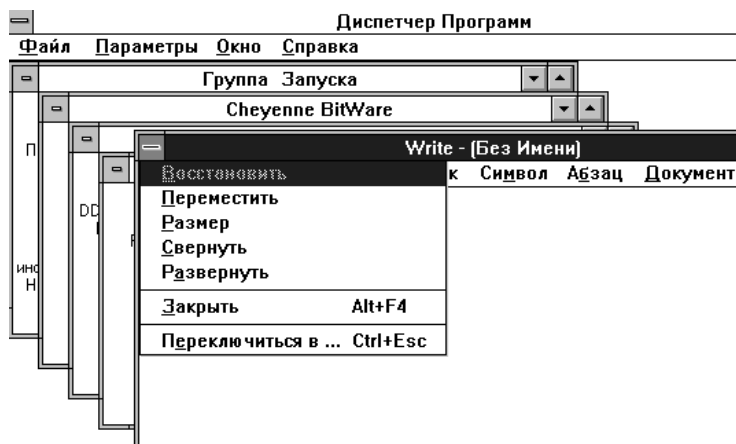


Рис. 2.6. Пример системного меню (редактор Write)

Меню содержит ряд директив, в большинстве своем предназначенных для изменения размеров и представления данного окна.

Посредством директив *Свернуть* (Minimize) и *Развернуть* (Maximize) можно менять представление окна точно так же, как с помощью размерных переключателей, расположенных в правом верхнем углу окна. Директива *Восстановить* (Restore) доступна только в случае представления окна в виде пиктограммы или в полноэкранном варианте. Она обеспечивает восстановление нормального представления окна из пиктограммы или из полноэкранного представления.

Воспользовавшись директивами *Restore* и *Maximize* из этого меню, можно превратить пиктограмму в нормальное или полноэкранное окно соответственно.

Директивы *Move* (Перемещение, Перетаскивание, Транспортировка, Буксировка) и *Size* (Размер) позволяют с помощью клавиш управления курсором менять соответственно положение и размер окна. Если щелкнуть на директиве *Move*, то курсор примет вид четырехсторонней стрелки. Теперь с помощью клавиш управления курсором можно перемещать окно. Для закрепления его на новой позиции следует нажать <Enter>.

Аналогичным образом производится изменение размеров по директиве *Size*. После щелчка на ней курсор превращается в четырехстороннюю стрелку. Следует выбрать ту кромку окна, которую нужно переместить для изменения его размера. Выбор осуществляется нажатием соответствующей клавиши управления курсором. После этого курсор превращается в расположенную на выбранной кромке двунаправленную стрелку, показывающую возможные направления ее перемещения. Клавишами управления курсором можно перемещать кромку окна, а зафиксировать ее на новом месте можно с помощью клавиши *<Enter>*.

Директива *Заккрыть (Close)* закрывает окно и завершает работу с данным приложением (в данном случае с *Write*).

Самой нижней в системном меню является директива *Переключиться в... (Switch To...)*, позволяющая переключиться на другие активные в данный момент задачи или на Менеджер Программ.

Диспетчер Программ

Диспетчер Программ *Windows* (ДП, *Program Manager*) предназначен для запуска и завершения программ *Windows*, переключения между программами, имеет меню, содержащее четыре пункта: *Файл*, *Параметры*, *Окно*, *Справка* (см. рис. 2.5). Меню — это список выполняемых команд или действий.

<F1> — вызов справочной системы, получение помощи в *Windows*. Для лучшей структуризации имеющихся программ ДП позволяет объединять программы в группы.

Как правило, ДП используется в качестве оболочки, т. е. программы, запускаемой сразу после старта *Windows*. В этом случае при выходе из ДП происходит и выход из *Windows*. ДП может быть свернут (минимизирован) в пиктограмму на экране. Для разворачивания ДП в виде окна надо дважды щелкнуть по его пиктограмме мышью или щелкнуть один раз и указать пункт *Развернуть*.

Функции ДП. С помощью ДП можно организовать приложения и файлы по смысловым программным группам. ДП выполняет следующие функции с помощью директив в пункте меню *Файл*:

1. Упорядочивание окон и пиктограмм.
2. Изменение свойств программных групп и программных элементов.
3. Копирование и перемещение программных элементов.

4. Создание и удаление программных групп и программных элементов.

С помощью пункта меню **Окно** можно выбрать и открыть любое окно из предложенного списка окон, в том числе и Дополнительные окна. Если программные группы расположены на рабочем столе Диспетчера Программ хаотично, то их можно упорядочить, используя меню **Окно**, **Упорядочить**. Аналогично поступают, если программные элементы расположены в окне беспорядочно.

Программы, которые запускаются ДП, отображаются в его окне в виде пиктограмм, которые объединены в программные группы. Главное окно ДП называется Рабочим столом Windows (DeskTop). Программные группы представлены внутри ДП в виде окон (они свернуты в пиктограммы, которые все одинаковы и отличаются только подписями). Программные элементы, находящиеся внутри окон (программных групп), имеют свои оригинальные пиктограммы, по которым (а также по подписям) можно найти необходимую программу, которая запускается двойным щелчком мыши по пиктограмме.

Имеется три типа окон:

1. Окно программы (окно Диспетчера Программ — Рабочий стол), содержит выполняемую программу.
2. Окно документа — это окно программной группы, содержит значки (иконки) программных элементов.
3. Окна диалогов — в них выводятся запросы (типа Да, Нет).

Можно открыть несколько окон программных групп на Рабочем столе. Для перемещения окна надо, ухватившись мышью за его заголовок, перетащить его на новое место. Аналогично мышью перетаскивают на новое место программную группу и программный элемент.

Для создания Программной группы необходимо указать в меню мышью пункт **Файл\Создать\Группа Программная**, ввести имя группы на русском или английском языке, ОК.

Для создания Программного элемента внутри программной группы необходимо указать в меню мышью пункт **Файл\Создать\Программный элемент**, ввести описание или согласиться с тем, что Программному элементу будет дано имя исполнимого файла, пролистать, найти на диске имя исполнимого файла (* .exe, * .com, * .bat, * .pif), ОК.

Программы DOS получают пиктограмму MS DOS, программы Windows имеют обычно свою собственную пиктограмму.

Пиктограммы (иконки) — это небольшие значки (картинки), представляющие собой свернутые объекты, такие, как:

1. Документ (пиктограмма программной группы).
2. Программа (пиктограмма программного элемента).
3. Файл (для Диспетчера Файлов).
4. Директории (для Диспетчера Файлов).

Для смены пиктограммы надо указать в меню мышью пункт `Файл\Свойства\Изменить значок`, пролистать, выбрать библиотеку пиктограмм, которую содержит сам Диспетчер Программ, выбрать подходящую пиктограмму, ОК. В Windows есть большая библиотека пиктограмм, которая содержится в файле `moricons.dll`. Из нее тоже можно выбрать подходящую пиктограмму. Пиктограммы (иконки) также содержатся в графических файлах `*.ico`. Их создают с помощью графического редактора `Icon Editor` из комплекта `Norton Desk Top`.

Стандартные графические файлы иконок `*.ico` имеют размер 766 байт.

Для смены описания (имени элемента или группы) надо их выделить, взять пункт меню `Файл\Свойства`, стереть с помощью `<BackSpace>` старое имя и ввести новое, ОК.

Для удаления Программного элемента (или Программной группы) ее сперва выделяют мышью, затем меню `Файл\Удалить\ОК`. Можно также использовать клавишу `<Del>`. Основные Программные группы и Программные элементы удалять нельзя. На их восстановление уйдет много времени. Удаление Программного элемента не означает удаления самой программы с диска.

В пункте меню `Файл` есть подпункт `Выполнить`, где есть Командная строка. Если ввести в нее, как в командной строке DOS, команду, то можно, нажав на `<Enter>`, выполнить эту команду.

Подпункт `Открыть` в меню `Файл` позволяет запустить программу с выделенным программным элементом.

После инсталляции в Windows имеется несколько основных Программных групп: Главная, Группа Запуска (в ней должен быть только индикатор клавиатуры), Реквизиты, Приложения и Игры.

Многозадачность или одновременная работа нескольких приложений. Указанная возможность осуществляется с помощью Списка Задач.

Вызовом Списка Задач — `<Ctrl+Esc>` или кнопкой «-» системного меню Диспетчера Программ\Переключиться в... можно запускать по очереди разные загруженные приложения,

указывая их в Списке Задач (Переключиться в... или Завершить Задачу). Сворачивание открытого приложения выполняется нажатием <Alt+Tab>. При этом можно запустить другое приложение, реализовав режим многозадачности. Повторно нажать <Alt+Tab> — снова развернуть приложение. Количество одновременно открытых приложений ограничено объемом оперативной памяти. Однако приложения все равно работают не одновременно, а по очереди, хотя и загружены в ОЗУ. При выходе из Windows надо закрыть все активные приложения.

Диспетчер файлов

Диспетчер файлов (File Manager) в Windows 3.1 является вспомогательной программой, открывающей особые возможности работы и выдвигающей при этом иногда и особые требования. В повседневной работе с Windows и Windows-приложениями не всегда есть необходимость в Диспетчере файлов (рис. 2.7). Однако существует ряд задач, которые не могут быть достаточно эффективно решены без его помощи.

Функции. Среди многих полезных свойств Диспетчера файлов следует выделить следующие два:

- с помощью Диспетчера файлов в Windows-среде можно так организовать свою работу, что почти не придется вни-

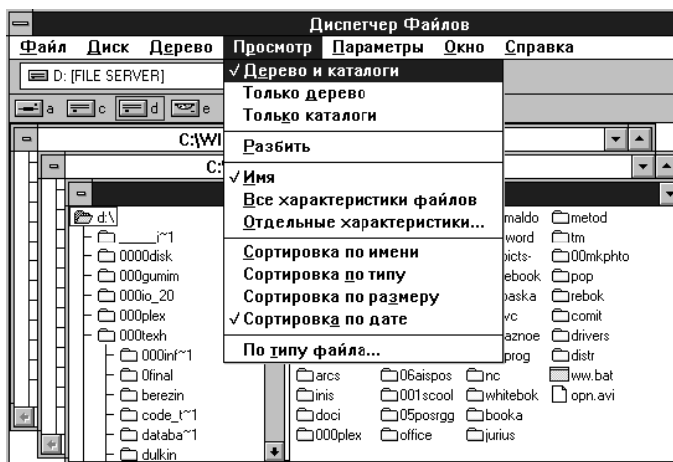


Рис. 2.7. Окно Диспетчера файлов (File Manager)

кать в подробности приложения (т. е. запоминать назначение его команд) и тем более беспокоиться о размещении результатов его работы, т. е. документов;

- Диспетчер Файлов является вспомогательной программой, обеспечивающей управление файловой системой DOS.

В обеих средах (DOS и Windows) принципы хранения информации на диске одинаковы: данные организуются в файлы, имеющие уникальные имена. Файл представляет собой организованную особым образом порцию информации, которой присвоено определенное имя. В основном существует два различных типа таких файлов, которые в Windows-среде различаются, в частности, и по расширениям, а именно:

- прикладные программы (приложения), т. е. файлы программ, исполняемые файлы, с помощью которых что-либо может быть обработано;
- документы, т. е. рабочие файлы, то, что обрабатывается с помощью прикладных программ: тексты, таблицы, записи данных, графические данные и др.

Диспетчер Файлов показывает, на каких дисководах (дисках), в каких директориях и какие именно файлы доступны. При этом он также условно показывает (по мере необходимости и возможности), к какому типу эти файлы относятся. С его помощью можно управлять файлами и обрабатывать их. При этом можно работать «документоориентированно», а также и обычным способом, как принято в DOS. Это совершенно непринципиально.

Еще одно полезное своеобразие Диспетчера Файлов заключается в том, что он так же, как и Диспетчер Программ, использует оконный интерфейс подчиненными окнами. Всегда существует «основное» прикладное окно Диспетчера Файлов, внутри которого может располагаться еще несколько подчиненных окон (но не более девяти).

Эти подокна показывают структуру и содержание директорий, поэтому они еще называются окнами директорий. Можно увеличить или уменьшить различные окна, расположить их каскадом или без перекрытий, но только в пределах прикладного окна Диспетчера файлов.

Пиктограмма файла-документа. Эта пиктограмма соответствует документам, точнее файлам, для обработки которых определено некоторое приложение. Доступ к такому приложению для обработки отмеченного такой пиктограммой файла можно полу-

чить двойным щелчком на пиктограмме (или имени файла с такой пиктограммой). Если в правой части окна ничего нет (например, по причине некорректной конфигурации Диспетчера Файлов), то следует обратиться к меню View (Вид) и установить там опцию Tree and Directory (Дерево и Список).

Приложения Windows 3.1

В стандартной поставке пакета Windows находится ряд приложений. Все они объединены в группу Accessories (реквизиты, аксессуары, инструменты) (рис. 2.8). Это небольшие по размеру и возможностям прикладные программы, составляющие «джентльменский набор» пользователя. Им далеко до профессиональных специализированных пакетов, но они иллюстрируют возможности оболочки и обеспечивают некоторый минимальный сервис.



Рис. 2.8. Стандартные приложения (Реквизиты, Accesories) Windows 3.1

Основные приложения Windows, входящие в его состав, расположены в программной группе Реквизиты (т. е. их программные элементы):

1. Графический редактор PaintBrush.
2. Текстовый редактор Write.
3. Простейший текстовый редактор txt-файлов Блокнот (NotePad).

4. Калькулятор (в стандартном виде и научный с элементарными функциями).
5. Часы (с цифровой или стрелочной индикацией).
6. Календарь (с будильником).
7. Карточка (простейшая база данных).
8. Таблица символов.
9. Раскладка клавиатуры.
10. Звукозапись (с микрофона в мультимедийных машинах).
11. Медиаплеер (для воспроизведения видео- и аудиозаписей в мультимедийных машинах).
12. Упаковщик объектов.

Помощь (справочная подсистема)

Оболочка Windows 3.1 располагает хорошо структурированной справочной подсистемой с большим объемом справочных текстов. Иногда эту подсистему называют подсистемой помощи (Help) по имени директивы, открывающей доступ к ее ресурсам. Справочная подсистема содержит информацию по всем компонентам системы, по каждому интегрированному в нее Windows-приложению. Кроме того, доступна контекстная справочная информация, т. е. информация о текущей ситуации в системе — ее можно вызвать в любой момент работы с оболочкой.

Принцип построения справочной подсистемы позволяет соблюдать единообразие в действиях при получении справки в различных ситуациях. Сама справочная подсистема содержит специальную справку о методах работы с ней. Доступ к этой справке можно получить, если в операционном меню Help выбрать директиву How to Use Help (Как пользоваться справкой) или, находясь в справочном окне, нажать функциональную клавишу <F1>.

Вызвать справку можно различными способами.

Наиболее удобным является так называемый контекстный вызов. Под этим термином кроется весьма удобное свойство справочной подсистемы — ее способность в момент вызова определять, какая именно справка нужна в данной ситуации. Определив, подсистема находит на диске требуемый файл, загружает его и выдает на экран подобранную справку.

После загрузки справки в справочное окно дальнейшая работа со справочной подсистемой выглядит во всех трех случаях одинаково.

Ситуационная справка. Ситуационная или контекстная справка вызывается с помощью клавиатуры. Обычно для этих целей используется функциональная клавиша <F1>. В некоторых случаях дальнейшую (более детальную) информацию можно получить, нажав <Shift+F1>. Справка, вызываемая в диалоговом окне щелчком на кнопке Help, также является контекстной (рис. 2.9).

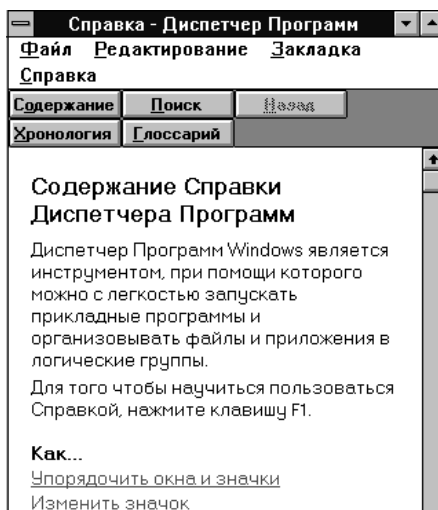


Рис. 2.9. Пример ситуационной справки

Если нажать на функциональную клавишу <F1> в момент, когда активен Диспетчер Программ, на экране появляется окно с заголовком Program Manager Help (Справка по Диспетчеру Программ). В данный момент в нем выведено Содержание справки по Диспетчеру Программ (Contents for Program Manager Help).

Windows for Workgroups 3.11

Последующая версия Microsoft Windows 3.11 была названа Windows for Workgroups 3.11. Основным отличием ее от версии Windows 3.1 является то, что в программный пакет интегрированы сетевые драйверы, позволяющие использовать его не только на отдельно стоящем ПК, но и в сети. Кроме того, в операционную среду включено несколько новых программ, значительно

изменен Диспетчер Файлов — одно из слабых мест предыдущих версий, расширен Диспетчер Печати и встроен факс. Разработан новый офисный пакет, включающий в себя текстовый редактор Word, электронные таблицы Excel, редактор формул Equation и ряд других возможностей. Внешний вид интерфейса Windows for Workgroups 3.11 представлен на рис. 2.10 и практически не отличается от Windows 3.1.

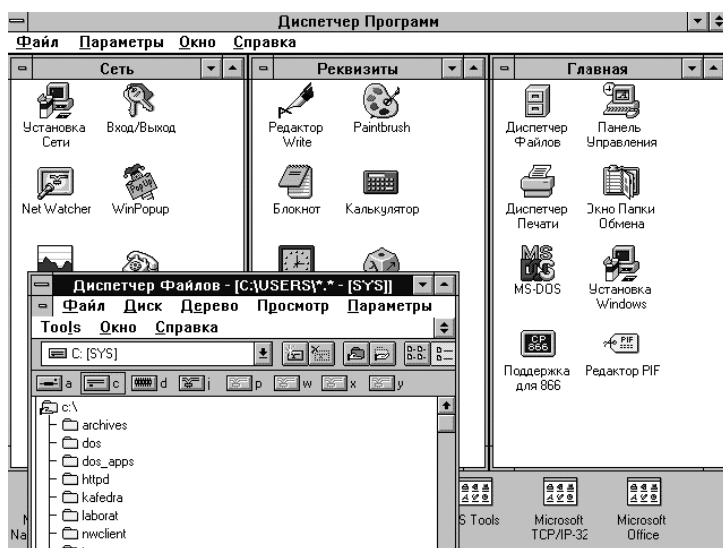


Рис. 2.10. Интерфейс операционной системы Windows for Workgroups 3.11

Само название системы «Windows для рабочих групп» означает то, что программа предназначена для работы на нескольких ПК равноправных пользователей, объединенных в рабочие группы.

В Windows for Workgroups дополнительно введен ряд сетевых функций. Наиболее заметные из них — средства организации одноранговых сетей, которые позволяют вам пользоваться диском или каталогом совместно с другими пользователями либо с помощью встроенных сетевых функций Windows, либо через более крупную сеть на основе сервера, такую, как NetWare фирмы Novell или Windows NT.

Кроме того, в Windows for Workgroups добавлен клиент электронной почты Microsoft Mail и планировщик Scheduler.

Для пользователей, не подключенных к сети, большое преимущество Windows for Workgroups состояло в применении 32-разрядного виртуального драйвера устройств (VxD) для файловой системы. Он может значительно повысить производительность многих накопителей с интерфейсом IDE.

Краткие сведения об архитектуре Windows 3.x

В соответствии с архитектурой Windows все прикладные программы и системный код размещаются в едином адресном пространстве. Это означает, что недоработанная прикладная программа, содержащая ошибки, может испортить области памяти, которые используются операционной средой или другой прикладной программой. Результатом будет весьма неприятная ошибка общего нарушения защиты (General Protection Fault). Иногда Windows с честью выходит из положения, восстанавливая свою работоспособность, но чаще всего это ей не удается.

В своей основе Windows 3.x — 16-разрядная операционная система, поэтому для программ память представляется состоящей из 64-Кбайт сегментов, а все данные в своей основе 16-разрядные. Такая система может оказаться менее эффективной по сравнению с 32-разрядной адресацией при работе с большими массивами данных. Еще одно следствие 16-разрядной базы этой ОС — ограниченность системных ресурсов. В Windows 3.x для хранения таких структур, как дескрипторы файлов прикладных программ, выделяется лишь небольшой блок памяти в других адресах. После того как эти области памяти заполнятся, Windows не может загрузить новые прикладные программы, даже если в ее распоряжении остается вполне достаточно памяти в других адресах (рис. 2.11).

В основе организации Windows 3.x лежит 16-разрядная архитектура. Ее ядро, большинство важнейших компонентов и собственные прикладные программы представляют собой 16-разрядные коды. (Ее редко используемый интерфейс Win32 API дает возможность выполнять 32-разрядные прикладные программы, но не позволяет работать с несколькими потоками.)

Все собственные прикладные программы Windows 3.x и все ее системные библиотеки DLL отображаются в общее сегментированное виртуальное адресное пространство размером 4 Гбайт. Все эти компоненты видимы (и часто доступны на уровне записи) друг для друга. В нижней части этого адресного пространства,

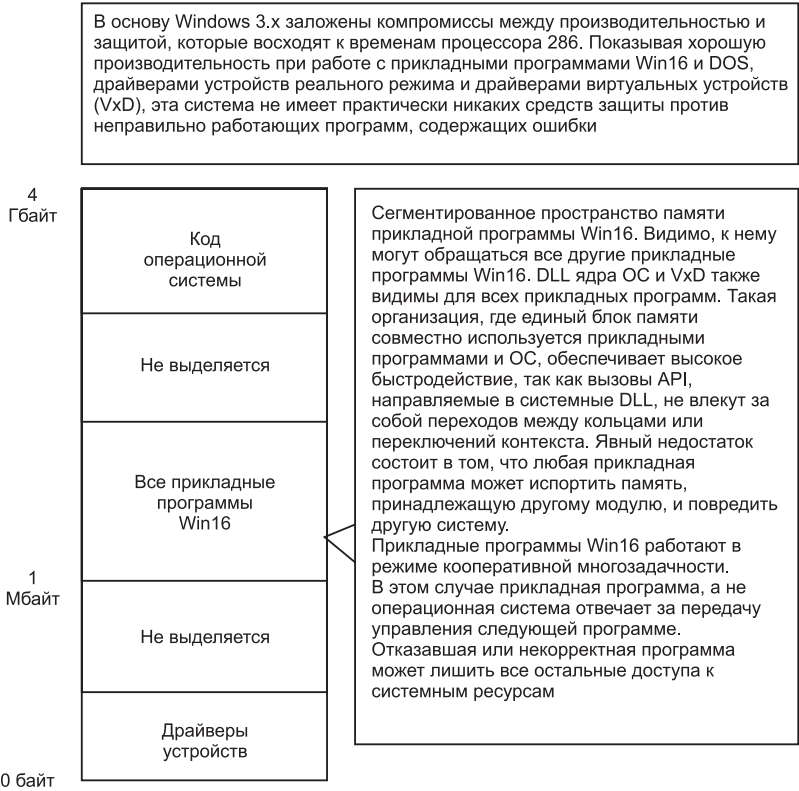


Рис. 2.11. Модель памяти Windows 3.x

обычно ниже метки 1 Мбайт, размещаются драйверы устройств реального режима, обеспечивающие взаимодействие с периферийными подсистемами, такими, как видеоплаты или принтеры. В Windows 3.11 драйверы VxD файловой системы используются для отыскания маршрута доступа к диску в защищенном режиме.

Windows 3.1x одновременно выполняет несколько прикладных программ с помощью простого механизма планирования, называемого кооперативной многозадачностью. В этой системе каждая прикладная программа должна добровольно уступить управление, когда, проверив свою очередь сообщений, она обнаруживает, что та пуста. Но если прикладная программа не проверит свою очередь сообщений либо по причине занятости, либо вследствие зависания, то другие прикладные программы лишатся доступа к совместно используемым ресурсам.

2.3. Операционная система OS/2



Разработка OS/2 начиналась совместно IBM и Microsoft (хотя большую часть работы должна была выполнить Microsoft). Изначально OS/2 была задумана как замена DOS, поскольку уже тогда было ясно, что DOS с ее ограничениями по памяти и возможностями файловой системы не может использовать вычислительные мощности появившихся на рынке компьютеров, в частности, вышедшей ранее серии IBM PS/2 (Personal System/2).

В конце 1994 г. IBM выпустила третью главную версию OS/2, которую назвала OS/2 Warp 3 (warp — основа).

Операционная система OS/2, будучи полноправной многозадачной операционной системой со своими оригинальным графическим пользовательским и программным интерфейсами, сохраняет совместимость с MS DOS, PC DOS и Microsoft Windows (начиная с версии WARP 3.0).

Общие характеристики OS/2

Многозадачность. В ОС OS/2 реализована возможность запуска двух или более программ одновременно, а планировщик задач определяет, какой из этих задач предоставить время процессора.

Многозадачная и многопоточная архитектура ОС OS/2 обеспечивает прочную платформу сетевым клиентам. OS/2 поддерживает прикладные программы отдельно от функций сети, таким образом, возникновение проблемы с программным обеспечением в одной области памяти не ведет к зависанию всей операционной системы и прерыванию работы. Это давало ОС OS/2 значительное преимущество перед Windows 3.1 и Windows for Workgroups 3.11, так как эти продукты не поддерживали технологию Intel Virtual Machine Technology, которая надежно управляет прикладными программами.

Сетевые возможности. Операционная система OS/2 поддерживает полный набор протоколов для работы как в локальной сети, так и в глобальной сети Internet. Это такие протоколы, как NetBIOS, IPX/SPX, TCP/IP и др.

Совместимость. ОС OS/2 позволяет запускать программы MS DOS и Windows посредством «блока совместимости» или

«блока реального режима». Программы MS DOS работают в реальном режиме (режим 80286, который эмулирует 8088 и 8086), тогда как программы, написанные специально для среды OS/2, работают в защищенном режиме, поскольку в этом режиме память и ресурсы одной программы защищены от других программ. Большинство программ, работающих под управлением MS DOS, могут работать в блоке совместимости, но есть и такие, которые не работают.

В основном все приложения Windows под OS/2 выполняются нормально. Однако некоторые Windows-программы используют так называемые виртуальные драйверы устройств (VxD). Эти виртуальные драйверы устройств представляют собой работающие в защищенном режиме 32-битовые динамически компоуемые библиотеки, которые управляют разделением системных ресурсов между несколькими прикладными программами. Операционная система OS/2 не осуществляет полную поддержку виртуальных драйверов Windows. Именно поэтому ОС OS/2 не позволяет использовать сетевые возможности Windows for Workgroups 3.11.

Интерфейс. Фирма IBM вместе с операционной системой OS/2 выпустила свой вариант графического интерфейса пользователя — Presentation Manager. Это развитый объектно-ориентированный программный интерфейс с применением техники drag-and-drop при выполнении операций копирования, удаления, печати, а также некоторых других. Перечни свойств объектов легко доступны в меню, вызываемых щелчком правой клавиши мыши. Имеется специальная панель для размещения часто используемых документов или прикладных программ. Все графические и программные примитивы представляются в ней в виде объектов (память, дисплей, принтер, папка, звуковая карта, дисковод).

В сентябре 1996 г. была выпущена версия OS/2 Warp 4.0 (под кодовым именем Merlin). Новые изменения и улучшения превратили Merlin в передового для того времени представителя операционных систем семейства OS/2 (рис. 2.12):

- встроенные возможности по управлению голосом и средства голосового набора позволили использовать голосовые команды при работе с системой (открыть веб-страницу, удалить объект и т. д.) и осуществлять голосовой ввод текста;
- графический интерфейс пользователя был кардинально переработан и улучшен. В дополнение к LaunchPad появился новый элемент WarpCenter, еще более упрощавший работу;



Рис. 2.12. Интерфейс OS/2 Warp (Merlin)

- встроенные средства для работы в Internet также значительно изменены и улучшены.
- появляется встроенная виртуальная Java-машина, благодаря этому OS/2 приобрела способность запускать Java-приложения, обеспечивая хорошую производительность.

Внутренняя организация OS/2 Warp

На рис. 2.13 показана структура операционной системы OS/2 Warp 3.0. В OS/2 имеется несколько видов виртуальных машин для прикладных программ. Собственные 32- и 16-разрядные программы OS/2 выполняются на отдельных виртуальных машинах в режиме вытесняющей многозадачности и могут общаться между собой с помощью средств DDE OS/2. Прикладные программы DOS и Win16 могут запускаться на отдельных виртуальных машинах в многозадачном режиме. При этом они поддерживают полноценные связи DDE и OLE 2.0 друг с другом и связи DDE с 32-разрядными программами OS/2. Кроме того, можно запустить

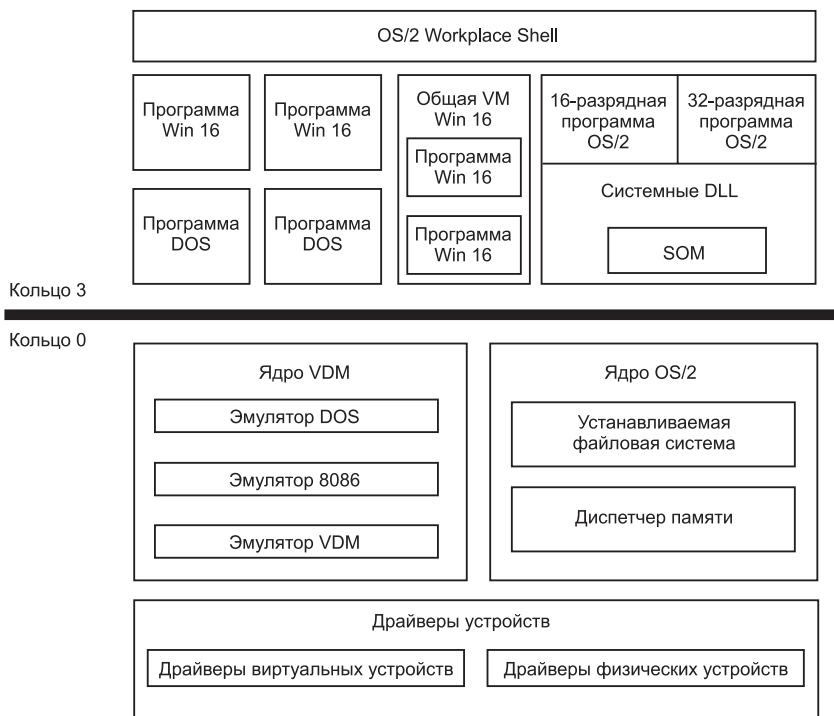


Рис. 2.13. Структура OS/2

несколько программ Win16 на общей виртуальной машине Win16, где они работают в режиме невытесняющей многозадачности. Разнообразные сервисные функции API OS/2, в том числе SOM (модель системных объектов), обеспечиваются с помощью системных динамических библиотек DLL, к которым можно обращаться без требующих затрат времени переходов между кольцами защиты. Ядро OS/2 предоставляет многие базовые сервисные функции API, обеспечивает поддержку файловой системы, управление памятью и имеет диспетчер аппаратных прерываний. В ядре виртуальных DOS-машин (VDM-ядре) осуществляется эмуляция DOS и процессора 8086, а также управление VDM. Драйверы виртуальных устройств обеспечивают уровень аппаратной абстракции. Драйверы физических устройств напрямую взаимодействуют с аппаратурой.

Первые версии OS/2 не оказали значительного влияния на рынок. Версия OS/2 1.0, выпущенная в 1987 г., содержала боль-

шинство технических свойств, необходимых для многозадачной ОС. Однако у нее не было менеджера графического представления (presentation manager — PM), а также отсутствовали драйверы для многих популярных принтеров и других устройств. Версия OS/2 1.1, появившаяся в 1989 г., включала рудиментарную версию PM, которая, наконец, делала возможным использование графических приложений в нескольких окнах. Однако в этой версии PM не хватало многих свойств, которые присущи развитому графическому интерфейсу, кроме того, по-прежнему отсутствовали многие драйверы принтеров. Выпущенная в 1990 г. версия 1.2 имела улучшенный PM, хотя он и не следовал общепринятым концепциям графического интерфейса. Появились драйверы для большинства принтеров и других периферийных устройств.

Однако версия 1.2 не была существенно лучше предыдущих версий и все еще предъявляла значительные требования к аппаратуре. К этому времени многие пользователи решили перейти на новую платформу Windows 3.0 или подождать, пока не появится что-нибудь принципиально лучшее. Продажи OS/2 по-прежнему были вялыми, и рынок не интересовался ею. Это объяснялось наличием у OS/2 ряда существенных недостатков:

- виртуальная машина DOS, которая должна была бы обладать способностью выполнять немодифицированные приложения DOS, с самого начала имела технические изъяны, поскольку была реализована на основе реального режима процессора. При этом требовался перезапуск процессора для переключения между реальным и защищенным режимами. Хотя эта операция и выполнялась очень быстро и незаметно для пользователя, она была сложной и вносила путаницу;
- Microsoft и IBM не смогли в полной мере реализовать концепцию виртуальной обработки в режиме I8086: в этом режиме DOS-приложения, которые непосредственно обращаются к аппаратным портам, перестают работать. В связи с этим не могли использоваться и популярные сетевые операционные системы на базе DOS;
- память в этом режиме использовалась нерационально — если пользователь конфигурировал OS/2 с возможностью DOS-совместимости, то 640 КБ памяти всегда выделялись для этих целей и не могли использоваться для задач OS/2;
- отсутствие возможности обмена данными между DOS- и OS/2-приложениями;

- в каждый момент времени могло выполняться только одно DOS-приложение, и это приложение не могло использовать расширенную память.

В результате для пользователей OS/2 многие популярные DOS-приложения оказались недоступными, а те, что были доступны, не могли вообще взаимодействовать со средой OS/2. Время показало, что для пользователей это обстоятельство оказалось весьма важным, так как многие отказались от покупки OS/2, оставаясь с проверенной, хотя и не очень совершенной DOS.

Файловая система HPFS

HPFS (High Performance File System, высокопроизводительная файловая система) была совместно разработана в 1989 г. корпорациями IBM и Microsoft, чтобы преодолеть недостатки FAT, к числу которых относятся:

- ограничения, налагаемые на размер файлов и дискового пространства;
- ограничение длины имени файла;
- фрагментация файлов, приводящая к снижению быстродействия системы и износу оборудования;
- непроизводительные затраты памяти, вызванные большими размерами кластеров;
- подверженность потерям данных.

Общие характеристики ФС. Проблема непроизводительных потерь дискового пространства связана с тем, что место на диске выделяется целыми блоками — кластерами, и в среднем половина одного (последнего) выделяемого кластера для каждого файла будет затрачиваться впустую. Чем больше размер раздела жесткого диска, тем больше объем минимальной неделимой области памяти, выделяемой файлу, тем больше потери.

Эти потери можно существенно сократить внедрением более эффективных файловых систем. Простой переход на HPFS, работающую в среде OS/2, позволяет вновь вернуться к первоначальному размеру выделяемого блока — 512 байт, причем для любых размеров диска. Размер вероятного выигрыша для диска емкостью 512 Мбайт, содержащего 8000 файлов, составит около 30 Мбайт. Этот выигрыш связан с тем, что на каждом файле в среднем теряется не 4096 байт (половина размера кластера в FAT для диска данной емкости), а всего 256 байт.

В OS/2 положение осложняется применяемым методом хранения расширенных атрибутов (extended attributes). В разделе FAT файл, содержащий единственный символ, занял бы целый кластер для размещения собственно файла и еще один кластер для расширенных атрибутов.

Так как расширенные атрибуты почти всегда имеют объем меньше 300 байт, размер теряемого впустую дискового пространства изменяется от примерно половины кластера при использовании малых разделов до львиной доли объема кластера при больших разделах. В сумме на каждом файле теряется примерно кластер.

Переход на HPFS позволяет сэкономить дисковое пространство. HPFS распределяет пространство, основываясь на физических 512-байтовых секторах, а не на кластерах, независимо от размера раздела. Система HPFS позволяет уменьшить и непроизводительные потери, так как в ней предусмотрено хранение до 300 байт расширенных атрибутов в F-узле файла, без захвата для этого дополнительного сектора.

Другая проблема связана с фрагментацией файлов, которая наиболее характерна для емких дисков с большим числом файлов. Фрагментация существенно сказывается на времени доступа к файлу. Другой негативный эффект фрагментации — повышенный износ диска. О серьезности этой проблемы говорит обилие утилит для дефрагментации дисков, использующих FAT.

Файловая система HPFS обеспечивает гораздо более низкий уровень фрагментации. Хотя избавиться полностью от нее не удастся, снижение производительности, возникающее по этой причине, почти незаметно для пользователя.

Рассмотрим физическую структуру ФС HPFS (рис. 2.14).

Управляющие блоки ФС. Первые 16 секторов (с 0 по 15) раздела HPFS составляют загрузочный блок (boot block). Эта область содержит метку диска и код начальной загрузки системы.

Сектор 16, известный под названием суперблок (super block), содержит общую информацию о файловой системе в целом: размер раздела, указатель на корневой каталог, счетчик элементов каталогов, номер версии HPFS, дату последней проверки и исправления раздела с помощью команды `chkdsk`, а также дату последнего выполнения процедуры дефрагментации раздела. Он также содержит указатели на список испорченных блоков на диске, таблицу дефектных секторов и список доступных секторов.

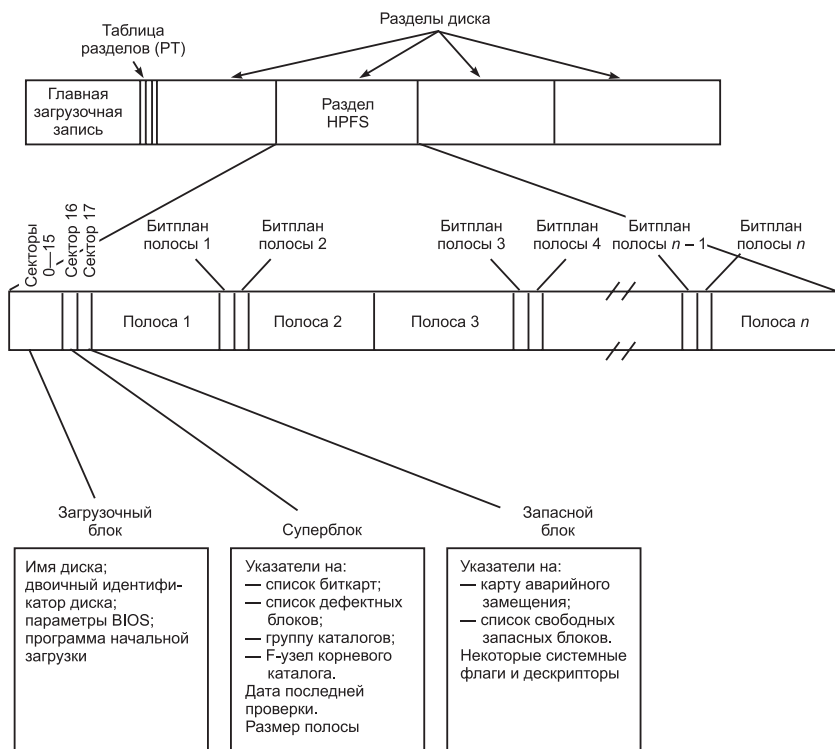


Рис. 2.14. Структура физического диска с разделом HPFS

Программа начальной загрузки находит файл `os2ldr`, считывает его в память и передает управление этой программе загрузки ОС, которая, в свою очередь, загружает с диска в память ядро `OS/2 — os2krnl`. И уже `os2krnl` с помощью сведений из файла `config.sys` загружает в память все остальные необходимые программные модули и блоки данных.

Сектор 17 носит название **запасной блок (spare block)**. Он содержит указатель на список секторов, которые можно использовать для «горячего» исправления ошибок, счетчик доступных секторов для «горячего» исправления ошибок, указатель на резерв свободных блоков, применяемых для управления деревьями каталогов, и информацию о языковых наборах символов. Система HPFS использует информацию о языковых наборах, чтобы дать возможность пересылать файлы, составленные на разных языках, даже в том случае, когда имена файлов содержат

уникальные для какого-либо языка символы. Запасной блок также содержит так называемый «грязный» флаг, который сообщает операционной системе о том, было ли завершение предыдущего сеанса работы нормальным, либо произошло в результате сбоя электропитания, либо файлы не были закрыты должным образом по какой-то другой причине. Если этот флаг обнаружен во время начальной загрузки, то операционная система автоматически запускает утилиту `chkdsk`, пытаясь обнаружить и исправить все ошибки, внесенные в файловую систему из-за неправильного выключения системы.

Все остальное дисковое пространство в HPFS разбито на части из смежных секторов — *полосы* (*band* — полоса, лента). Каждая полоса занимает на диске 8 Мбайт.

Каждая полоса имеет отдельную таблицу объемом 2 Кбайт (битовая карта распределения секторов, биткарта, битплан), в которой указывается, какие секторы полосы доступны, а какие заняты. Каждому сектору полосы данных соответствует один бит в ее битовой карте. Если бит установлен в «1», то сектор занят, если в «0» — свободен.

Чтобы максимально увеличить протяженность непрерывного пространства для размещения файлов, таблицы попеременно располагаются в начале и в конце полос (см. рис. 2.14). Этот метод позволяет файлам размером до 16 Мбайт (минус 4 Кбайта, отводимые для размещения таблицы) храниться в одной непрерывной области.

Принцип размещения файлов. Экстенты (*extent*) — фрагменты файла, располагающиеся в смежных секторах диска. Файл имеет по крайней мере один экстенст, если он не фрагментирован, а в противном случае — несколько экстенстов.

Для сокращения времени позиционирования головок чтения/записи жесткого диска система HPFS стремится:

- расположить файл в смежных блоках;
- если такой возможности нет, то разместить экстенсты фрагментированного файла как можно ближе друг к другу.

Для этого HPFS использует статистику, а также старается условно резервировать хотя бы 4 Кбайта места в конце файлов, которые растут. Если же часть этого пространства не потребовалась, то после закрытия файла она высвобождается для дальнейшего использования. Если же файл не может быть увеличен без нарушения его непрерывности, HPFS снова резервирует 4 Кбайт смежных блоков как можно ближе к основной части файла.

Файлы, состоящие из двух-трех экстенгов, практически не снижают производительность HPFS, так как эта файловая система следит за тем, чтобы области данных, принадлежащие одному и тому же файлу, располагались как можно ближе друг к другу.

Каждому файлу и каталогу диска соответствует файл о в ы й узел (F-node, F-узел), который занимает один сектор, всегда располагается поблизости от управляемого им файла или каталога (обычно — непосредственно перед файлом или каталогом) и содержит информацию о расположении файла и некоторые актуальные данные (рис. 2.15):

- размер файла;
- первые 15 символов имени файла;
- статистику доступа к файлу;
- расширенные атрибуты файла;
- список прав доступа (или только часть этого списка, если он велик, причем, если расширенные атрибуты слишком велики для файлового узла, то в него записывается указатель на них);
- связанную информацию о расположении и подчинении файла и т. д.

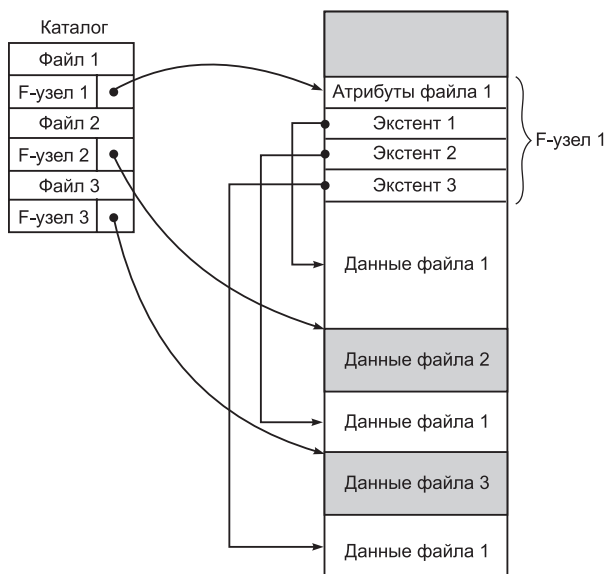


Рис. 2.15. Структура и размещение файловых узлов HPFS

Если файл непрерывен, то его размещение на диске описывается двумя 32-битовыми числами. Первое число представляет собой указатель на первый блок файла, а второе — длину экстенста (число следующих друг за другом блоков, принадлежащих файлу).

Если файл фрагментирован, то размещение его экстенстов описывается в файловом узле дополнительными парами 32-битовых чисел.

В файловом узле можно разместить информацию максимум о восьми экстенстах файла. Если файл имеет больше экстенстов, то в его файловый узел записывается указатель на блок размещения (allocation block), который может содержать до 40 указателей на экстенсты или, по аналогии с блоком дерева каталогов, на другие блоки размещения.

Таким образом, двухуровневая структура блоков размещения может хранить информацию о 480 секторах, что позволяет работать с файлами размером до 7,68 Гбайт.

Структура и размещение каталогов. Для хранения каталогов используется полоса, находящаяся в центре диска, называемая directory band. Если она полностью заполнена, HPFS начинает располагать каталоги файлов в других полосах.

Расположение этой информационной структуры в середине диска значительно сокращает среднее время позиционирования головок чтения/записи — для перемещения головок чтения/записи от произвольной дорожки диска к центральной требуется в 2 раза меньше времени, чем для перемещения к внешней дорожке (место размещения корневого каталога в случае файловой системы FAT). Уже только одно это обеспечивает более высокую производительность файловой системы HPFS по сравнению с FAT. Аналогичное замечание справедливо и для NTFS, которая тоже располагает свой master file table в начале дискового пространства, а не в его середине.

Однако существенно больший (по сравнению с размещением Directory Band в середине логического диска) вклад в производительность HPFS дает использование метода сбалансированных двоичных деревьев для хранения и поиска информации о местонахождении файлов (в то время как в файловой системе FAT каталог имеет линейную структуру, специальным образом не упорядоченную, поэтому при поиске файла требуется последовательно просматривать его с самого начала).

В HPFS структура каталога представляет собой сбалансированное дерево с записями, расположенными в алфавитном порядке.

Каждая запись, входящая в состав дерева, содержит:

- атрибуты файла;
- указатель на соответствующий файловый узел;
- информацию о времени и дате создания файла, времени и дате последнего обновления и обращения;
- размеры области данных, содержащей расширенные атрибуты;
- счетчик обращений к файлу;
- длину имени файла;
- само имя файла и др.

Файловая система HPFS при поиске файла в каталоге просматривает только необходимые ветви двоичного дерева (B-Tree). Такой метод во много раз эффективнее, чем последовательное чтение всех записей в каталоге, что имеет место в системе FAT.

Размер каждого из блоков, в терминах которых выделяются каталоги в текущей реализации HPFS, равен 2 Кбайт. Размер записи, описывающей файл, зависит от размера имени файла. Если имя занимает 13 байтов (для формата 8.3), то блок из 2 Кбайт вмещает до 40 описателей файлов. Блоки связаны друг с другом посредством списка.

Следует учесть, что многие современные контроллеры дисков могут читать за одно обращение сразу несколько секторов. Применяемая в HPFS схема значительно повышает шансы использовать эту возможность, при этом происходит еще большее уменьшение числа требуемых операций взаимодействия между программой, файловой системой, драйвером дискового устройства и физическим диском.

HPFS имеет и другие оптимизирующие функции. Так, при открытии или создании файла интеллектуальный алгоритм выделяет наиболее подходящую полосу. Программный интерфейс, используемый для создания файла, позволяет программисту сообщить операционной системе предполагаемый размер файла. С помощью этой информации HPFS может заранее выбрать для размещения файла полосу, имеющую непрерывную область наибольшего размера. Именно поэтому HPFS наиболее эффективно работает в больших разделах — большее число полос предоставляет большие возможности выбора.

Надежность ФС. HPFS обладает повышенной отказоустойчивостью по сравнению с FAT. Если на диске с FAT оказалась разрушенной таблица распределения файлов, то скорее всего окажутся потерянными все данные, которые находятся вне кор-

нового каталога. Поскольку в системе HPFS вместо таблицы размещения файлов применяется битовый массив, который содержит флаг, помечающий используемые секторы, то если область битового массива будет разрушена, пользователь этого не заметит, даже если это случится во время работы системы. F-узел файла также содержит информацию о размещении каждого файла. Поэтому область битового массива может быть восстановлена после поиска этой информации в F-узлах. Пользователь не увидит даже предупреждения — восстановление выполняется автоматически. Этот процесс может быть запущен и с помощью утилиты `chkdsk`, которая сравнивает битовый массив с информацией для файла о принадлежащих ему секторах. Если при чтении битового массива обнаруживается ошибка, то создается новый битовый массив.

В системе FAT при порче каталогов теряются указатели на начало цепочки кластеров каждого файла. Можно соединить отдельные кластеры в файл, но многое придется делать вручную. Так как утилиты, подобные `chkdsk`, «не знают» имени файлов, то для того, чтобы восстановить их старые имена, придется загружать файлы в текстовый редактор и пытаться определить, что они из себя представляют.

При работе с HPFS в случае потери каталога у каждого файла из этого каталога теряется лишь дата последней операции записи в файл и иных изменений, дата создания и длинное имя файла (символы, следующие за первыми пятнадцатью). Элемент каталога — это всего лишь указатель на F-узел. В F-узле хранятся первые 15 символов имени файла (плюс информация о том, имелись ли в имени файла другие символы, кроме первых 15) и прочая информация, нужная для доступа к файлу. Утилиты восстановления могут впоследствии найти в F-узле сведения о том или ином файле. Эта избыточность, обеспечиваемая каталогом и F-узлами, значительно увеличивает шансы на восстановление данных.

2.4. Операционные системы Windows 95/98/Me



Первые версии Windows 3.x нельзя назвать полноценными операционными системами, так как для их работы обязательно было наличие активной копии MS DOS. Таким образом, Windows 3.x являлась «оболочкой», по-

средником пользователя и операционной системы, облегчая процесс общения между ними.

В качестве развития серии Windows были выпущены две параллельные ветви операционных систем с графическим интерфейсом:

- Windows 95/98/Me;
- Windows NT/2000/XP.

Системы, внешне весьма схожие интерфейсами и предоставляемыми пользователю возможностями, коренным образом различаются по назначению и администрированию. Основные отличия заключаются в том, что предназначенная для использования в быту и в малых офисах серия 95/98 базируется на принципе Plug&Play подключения новых устройств (автоматический поиск и установка драйвера вновь появившегося устройства) и не требует серьезного администрирования, в то время как серия NT ориентирована на сетевое использование в больших организациях и требует точного конфигурирования и постоянного администрирования. Остальные особенности этих систем будут рассмотрены в свое время.

Следует отметить, что системы Windows 2000, а особенно Windows XP в существенной степени, интегрируют свойства первоначальных версий (Windows 95/98 и NT), как возможность точного администрирования и защиты системы, так и способность к автоконфигурированию.

Объектно-ориентированный подход

При создании Windows 95/98 в полной мере реализован объектно-ориентированный подход. Понятие «объектно-ориентированный» возникло в программировании сравнительно недавно. Когда вычислительная мощность машин была невысока, о создании объектно-ориентированных систем не могло быть и речи. Основой всего был программный код. Программисты записывали последовательности команд для выполнения тех или иных действий над данными, которые оформлялись в модули и процедуры. Для работы с каждым объектом создавалась своя процедура.

Объекты, их свойства и методы. Постепенно с увеличением производительности вычислительных систем процедурный подход начал заменяться объектным. На первое место выдвинулся объект, а не код, который его обрабатывает. На уровне пользова-

теля объектный подход выражается в том, что интерфейс представляет собой подобие реального мира, а работа с машиной сводится к действиям с привычными объектами. Так, папки можно открыть, убрать в портфель, документы — просмотреть, исправить, переложить с одного места на другое, выбросить в корзину, факс или письмо — отправить адресату и т. д. Понятие объекта оказалось настолько широким, что до сих пор не получило строгого определения.

Объект, как и в реальном мире, может обладать различными свойствами. Программист или пользователь может изменять не все свойства объектов, а только некоторые из них. Можно изменить имя объекта, но нельзя изменить объем свободного места на диске, который также является его свойством. Свойства первого типа в языках программирования носят название read/write (для чтения и записи), а свойства второго — read only (только для чтения).

Метод — это способ воздействия на объект. Методы позволяют создавать и удалять объекты, а также изменять их свойства. Например, для того чтобы нарисовать на экране точку, линию или плоскую фигуру, составляются разные последовательности кодов или программы. Пользователь, однако, применяет для отображения этих объектов один метод Draw(), который содержит коды для отображения всех объектов, с которыми он работает. За такое удобство приходится платить тем, что объектно-ориентированные системы могут работать только на достаточно мощных вычислительных установках.

Windows 95/98, основные особенности

С точки зрения базовой архитектуры Windows 95/98 — 32-разрядная, многопоточная операционная система с вытесняющей многозадачностью, что ставит ее в один ряд с такими соперниками, как OS/2, Unix и Windows NT. В ее среде могут выполняться собственные 32-разрядные прикладные программы, написанные в соответствии со спецификацией Win32 API (почти идентичный вариант этого интерфейса реализован в Windows NT). Собственные прикладные программы Windows 95/98 используют неструктурированное 32-разрядное адресное пространство, что делает их потенциально более быстродействующими при обработке больших массивов данных.

Компоненты ядра Windows 95/98. Ядро Windows 95/98 состоит из трех компонент:

- User управляет вводом с клавиатуры, от мыши и других координатных устройств, а также выводом через интерфейс пользователя. В Windows 95/98 используется модель асинхронного ввода;
- Kernel обеспечивает базовые функциональные возможности операционной системы (поддержку файлового ввода-вывода, управление виртуальной памятью, планирование задач), загружает exe- и dll-файлы при запуске программы, обрабатывает исключения, обеспечивает взаимодействие 16- и 32-разрядного кодов;
- GDI — графическая система, управляющая всеми объектами на экране дисплея и поддерживающая графический вывод на принтер и другие устройства.

В Windows 95/98 каждая 32-разрядная прикладная программа выполняется в собственном адресном пространстве, но все они совместно используют один и тот же 32-разрядный системный код. Неправильно написанная 32-разрядная программа все еще может привести к аварийному сбою всей системы. Все 16-разрядные программы Windows 95/98 разделяют общее адресное пространство, поэтому они столь уязвимы друг для друга, как и в среде Windows 3.1. В практической работе Windows 95/98 являлась более устойчивой средой, чем предшествующие версии Windows. Очевидны крупные изменения в пользовательском интерфейсе Windows 95/98. Используется кнопка Start для запуска прикладных программ самих по себе или через документы, с которыми программы связаны. После запуска программ их пиктограммы появляются на линейке заданий, обычно размещаемой в нижней части экрана. Щелчок на любой кнопке линейки заданий вызывает переключение на соответствующую программу. Это самый интуитивно понятный способ переключения задач из всех когда-либо существовавших.

Модули Program Manager и File Manager уступили место образу рабочего стола, на котором файлы пользователя показаны в виде пиктограмм, помещенных в так называемые папки. Более сложные функции по управлению файлами Windows 95/98 поручены утилите Проводник (Explorer), по существу заменившей File Manager, которая показывает древовидную диаграмму файловой структуры компьютера и его сетевого окружения. Благодаря расширению файловой системы FAT имена файлов не

ограничены, как раньше, восемью символами плюс состоящим из трех букв расширением; а можно использовать имена длиной до 255 символов.

Среди прочих благоприятных изменений в пользовательском интерфейсе — анимационные пиктограммы и диалоговые окна с закладками. В целом новый интерфейс представляет собой существенное улучшение по сравнению с Windows 3.1, хотя прежним пользователям могло потребоваться определенное время, чтобы привыкнуть к нему.

Метафора Рабочего стола (Desktop), основанная на приращении папок, и длинные имена файлов не изобретены создателями Windows 95/98, а в течение длительного времени были составной частью различных пользовательских интерфейсов, начиная с Macintosh и кончая Workplace Shell операционной системы OS/2.

Увеличилось число и повысилось качество поставляемых вместе с Windows 95/98 стандартных вспомогательных программ — от традиционного калькулятора и игр до мощных инструментальных средств контроля состояния системы. Также очевидны значительные усовершенствования средств связи.

Важнейшими встроенными приложениями Windows 95/98 (как и последующих систем, см. табл. 2.3 и пр.) являлись:

 **Блокнот** (NotePad) — простейший текстовый редактор (notepad.exe);

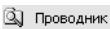
 **Калькулятор** — обычный и научный (calc.exe);

 **Paint** — графический редактор, аналог PaintBrush (pbrush.exe);

 **WordPad** — текстовый редактор (улучшенный аналог Write, write.exe);

 **Часы** (clock.exe);

 **Прогриватель** — медиаплеер (mplayer.exe) — служит для проигрывания видео- и аудиоклипов на мультимедийных компьютерах;

 **Проводник** — (Explorer, explorer.exe) — играет роль оболочки или менеджера файлов;

 Лазерный проигрыватель — (cdplayer.exe) — предназначен для проигрывания AudioCD;

 Дефрагментация диска — (defrag.exe) — используется для дефрагментации жесткого диска, которая повышает скорость загрузки и выполнения программ;

 Архивация данных — программа архивации предоставляет расширенные возможности архивации и восстановления данных, в том числе поддержку большего числа накопителей на магнитной ленте и самого современного оборудования;

Буфер обмена — (Clipboard, clipbrd.exe);

ScanDisk для Windows — (scandiskw.exe) — основной инструмент исправления ошибок в файловой системе Windows при сбоях. Проверка диска запускается автоматически после неверного выключения операционной системы. Программа проверки диска обнаруживает наиболее вероятные повреждения файлов и папок и выполняет исправление ошибок. Кроме того, пользователь имеет возможность выполнить проверку диска в любое время.

Имелся еще ряд других приложений Windows (набиратель телефонного номера, фонограф для записи и воспроизведения звука, эмулятор терминала, календарь, игры и др.).

Были расширены сетевые функциональные возможности. В состав Windows 95/98 был включен встроенный клиент для сетей NetWare 3.x, 4.x и для серверов Windows NT. Предусмотрены также средства для работы с протоколами IPX/SPX, NetBEUI, TCP/IP. Последний из перечисленных протоколов позволяет выполнять подключение к Internet. Windows 95/98 позволял непосредственно подсоединяться к другому компьютеру через кабель и располагает базовыми средствами для установления коммутируемых соединений через телефонные линии с сервером удаленного доступа Remote Access Server системы Windows NT, NetWare Connect или с коммутируемыми серверами компании Shiva. В состав Windows 95/98 также входил интерфейс прикладного программирования для телефонии (TAPI) фирмы Microsoft, обеспечивающий совместную работу машины с телефоном, регистрируя телефонные вызовы и выполняя функции автоответчика (прикладные программы для телефонии поставлялись независимыми фирмами).

Основные достоинства Windows 95/98:

- практически полная 32-битовая операционная система, что ускоряет работу многих программ по сравнению с 16-бито-

вой операционной системой MS DOS 6.22 и операционной оболочкой Windows 3.1;

- удобный графический многооконный интерфейс для пользователя (Рабочий стол — Desktop);
- возможность создания на Рабочем столе Ярлыков и Папок важнейших программ для их быстрого запуска. Внутри Папок могут находиться другие Папки и Ярлыки, что создает удобство в работе;
- запуск прикладных программ и возможность создания Меню при нажатии кнопки  (Start) в Панели задач;
- самонастраивающаяся система драйверов поддержки аппаратной части компьютера (технология «Plug and Play» — «Подключай и Работай»);
- реальная многозадачность (по сравнению с Windows 3.1);
- развитые сетевые функции, включая доступ к Internet;
- большой выбор прикладного программного обеспечения и поддержка большинства 16-битовых приложений;
- поддержка мультимедийных приложений;
- ориентация большинства производителей компьютерной техники и программного обеспечения на Windows 95/98.

Недостатки ОС Windows 95/98:

- высокие требования к аппаратной части компьютера (процессору, ОП, жесткому диску);
- недостаточная устойчивость в работе, особенно у русскоязычных версий. При сбоях в работе Windows 95/98 автоматически запускается программа Scandisk, которая, как правило, устраняет сбой и восстанавливает нормальную работу Windows 95/98;
- при выключении или перезагрузке компьютера с ОС Windows 95/98 надо обязательно использовать кнопку Пуск, Завершение работы, Выключить (Перезагрузить) компьютер, или клавиши <Alt+F4>, <Enter>. Это по сути дела есть парковка компьютера в Windows 95/98. При этом сохраняется текущая конфигурация и очищаются внутренние буфера. Только после этого пользователь имеет право выключить компьютер (на экране появится соответствующее сообщение). В противном случае возможны сбои и потеря важной информации;
- для обслуживания диска на компьютере с ОС Windows 95/98 следовало использовать только собственные средства Win-

dows 95/98 (ScanDisk и Defrag для Windows 95/98) или специальные утилиты Norton Utilities-9 для Windows 95/98 и не использовать старые утилиты (NU-6,7,8) Speedisk, Norton Disk Doctor, особенно для русскоязычной версии Windows 95/98, иначе можно было бы повредить файловую систему.

Основные отличия Windows 98



Средства Windows 98 облегчают работу с компьютером и расширяют возможности его использования.

Поддержка нескольких мониторов делает возможной использование нескольких мониторов для расширения рабочего стола, выполнения разных программ на разных мониторах, а также выполнение многоэкранных программ или игр. Например, студенты получают возможность открыть энциклопедию Microsoft Encarta на одном мониторе и печатать собственный отчет в Microsoft Word на втором мониторе.

Управление питанием. Выбор режима *Всегда Вкл* сокращает время запуска компьютера. При использовании средств управления питанием в режиме *Всегда Вкл* для запуска компьютера достаточно нескольких секунд. При этом все программы восстанавливаются в том состоянии, которое они имели на момент отключения. Кроме того, этот режим позволяет компьютеру работать даже тогда, когда он кажется выключенным. Пользователь получает возможность оставить все программы выполняющимися, загружать нужные Web-страницы, отправлять и получать электронную почту, архивировать жесткий диск или выполнять настройку операционной системы без необходимости находиться у компьютера. Кроме того, средства управления питанием позволяют перевести компьютер в режим ожидания (спящий режим) для сохранения ресурсов питания.

Шина USB. Универсальная последовательная шина (USB) облегчает использование компьютера за счет расширенных возможностей самонастраивающихся (plug-and-play) устройств. Универсальный стандартный разъем позволяет добавлять устройства без необходимости перезагружать компьютер.

Повышение надежности. В Windows 98 надежность компьютера повышается за счет применения новых мастеров, служебных программ и ресурсов, обеспечивающих бесперебойную работу системы.

Проверка системных файлов — контроль наиболее важных файлов, обеспечивающих работу компьютера. Если эти файлы повреждены или перемещены, программа проверки системных файлов их восстанавливает.

Проверка реестра реализуется системной программой, позволяющей обнаруживать и устранять ошибки. При каждом запуске компьютера программа проверки реестра автоматически проверяет реестр на наличие несогласованности структуры данных.

Кроме того, программа проверки реестра каждый день выполняет резервирование реестра. Если обнаруживаются серьезные ошибки в реестре, он может быть восстановлен по резервной копии. Программа проверки реестра поддерживает до пяти сжатых архивных копий реестра, при которых компьютер успешно запускался. Если архив не удается обнаружить, программа проверки реестра исправляет ошибки реестра.

Программа установки автоматически запускает проверку реестра при каждом обновлении операционной системы компьютера. При установке Windows 98 программа проверки реестра исправляет большинство ошибок в реестре, даже тех, о которых было неизвестно пользователю.

Архивация данных. Программа архивации предоставляет расширенные возможности архивации и восстановления данных, в том числе поддержку большего числа накопителей на магнитной ленте и самого современного оборудования. Пользователям становится легче сохранять важные данные. Файлы с жесткого диска можно резервировать на гибких дисках, на магнитной ленте или на другом компьютере в сети. Если исходные файлы повреждены или потеряны, их можно восстановить из архива.

Быстрая операционная система. Windows 98 включает средства, позволяющие компьютеру работать быстрее по сравнению с Windows 95/98 без добавления нового оборудования. В состав Windows 98 входит ряд программ, совместное применение которых повышает производительность компьютера.

Мастера Windows. Разработанные Microsoft Мастера стали популярным средством для проведения пользователя через последовательные этапы сложных процедур. С помощью серии достаточно простых вопросов пакету удастся досконально выяснить, чего конкретно хочет пользователь, и выполнить соответствующую операцию.

Мастер обслуживания повышает производительность системы и позволяет быстрее выполнять программы, проверять жест-

кий диск на наличие ошибок и освобождать место на диске. Создав расписание для регулярного выполнения этих служебных программ, можно добиться максимальной производительности компьютера. Например, оставляя компьютер включенным на ночь — составить расписание для выполнения этих задач в указанное время каждую ночь, раз в неделю или с любым другим интервалом.

Мастер подключения к Internet позволяет зарегистрироваться для доступа к Internet и автоматически выполняет шаги по настройке программного обеспечения, необходимые для доступа к Internet.

Проверка диска запускается автоматически после неверного выключения операционной системы. Программа проверки диска обнаруживает наиболее вероятные повреждения файлов и папок и выполняет исправление ошибок. Кроме того, пользователь имеет возможность выполнить проверку диска в любое время.

Преобразование ФС диска в систему FAT32, которая является расширенной версией файловой системы FAT16, позволяет форматировать как один раздел диски с емкостью более 2 Гбайт. Преобразованные диски используют кластеры меньших размеров, чем на дисках FAT16, в результате чего повышается эффективность использования объема диска. В состав Windows 98 включена служебная программа преобразования диска с графическим интерфейсом, которая позволяет быстро и безопасно преобразовать диск из исходной системы FAT16 в систему FAT32.

Дефрагментация диска повышает скорость загрузки и выполнения программ. Быстрый запуск и выключение позволяют быстрее и эффективнее работать, играть и путешествовать по Internet.

Полное объединение с Web. Проводник Windows 98 и Internet Explorer позволяют объединить ресурсы Web в едином представлении.

Расширенные средства Web. Windows 98 делает наиболее продуктивным использование Web за счет применения всех возможностей компьютера к интерактивному содержимому Internet:

- автоматическое дополнение ранее вызывавшихся адресов Web по мере их ввода;
- улучшенные списки часто посещаемых Web-узлов;
- улучшенный журнал и возможности отслеживания посещаемых Web-узлов;

- поддержка всех основных стандартов Internet, в том числе ActiveX, Java и др.;
- повышенная производительность динамического языка HTML, что позволяет сделать Web-страницы более богатыми и интересными.

Общий доступ к подключению Internet. Windows 98, второй выпуск, предоставляет пользователям возможность общего доступа к подключению Internet для нескольких компьютеров домашней сети. При этом один компьютер имеет непосредственный доступ к подключению Internet, а запросы от остальных компьютеров домашней сети направляются в Internet через этот компьютер. Кроме того, общий доступ к подключению Internet позволяет организовать доступ пользователей Internet к Web-серверам, а также почтовым и игровым серверам, размещенным в домашней сети.

Рабочий стол «Active Desktop» делает возможной настройку рабочего стола, запуск программ, переключение между файлами и отслеживание последних мировых новостей за счет объединения Web и рабочего стола пользователя. Active Desktop позволяет преобразовывать элементы Web в элементы рабочего стола и обновлять их в любое время.

Электронная почта. Панель управления Windows 98 включает программу Outlook Express, которая предоставляет защищенные средства для личной электронной почты и подключения к группам новостей. Для запуска Outlook Express следует нажать кнопку Пуск и выбрать команды Программы и Outlook Express.

NetMeeting — программа, позволяет вести разговоры по цифровым каналам связи с родственниками, друзьями и деловыми партнерами по всему миру без больших расходов. Кроме того, NetMeeting делает возможной совместную работу группы пользователей любых приложений для Windows с помощью общей доски, текстовых сообщений и передачи файлов. Если на компьютере установлено необходимое оборудование, становятся возможными телеконференции в режиме реального времени.

Функции и состав ОС Windows 95/98

Управление процессами. В Windows 95/98 процесс — это либо виртуальная машина MS DOS, либо любое работающее приложение Windows. Каждый процесс может порождать множество потоков. *Поток* — это последовательность команд в пределах процес-

са. Системный планировщик процессов (часть диспетчера виртуальной машины) управляет именно потоками.

Организация файловой системы. Основной файловой системой является VFAT. Пересмотренная версия Windows 95/98 OSR2 (OEM Service Release 2) поддерживает файловую систему FAT32, обеспечивающую поддержку жестких дисков размером более 2 Гбайт и более эффективное распределение дисковой памяти благодаря тому, что размер кластера в ней всего 4 Кбайт.

Поддержка технологии Plug-&-Play (PNP). Технология предназначена для упрощения установки и конфигурирования новых устройств. Устройства, соответствующие данной технологии, обязаны «уметь» сообщать ОС о своем наличии и о требуемых для работы ресурсах. С другой стороны, ОС обязана «уметь» распознавать такие устройства и автоматически выделять требуемые ресурсы.

В Windows 95/98 основным средством поддержки PNP является диспетчер конфигурации. Он осуществляет идентификацию всех устройств, загружает необходимые драйверы и с целью выделения ресурсов обращается к арбитрам ресурсов.

Управление ресурсами включает в себя распределение памяти, процессорного времени, управление доступом к устройствам ввода-вывода и т. д. Эти функции выполняются диспетчером виртуальной машины и ядром системы (рис. 2.16).

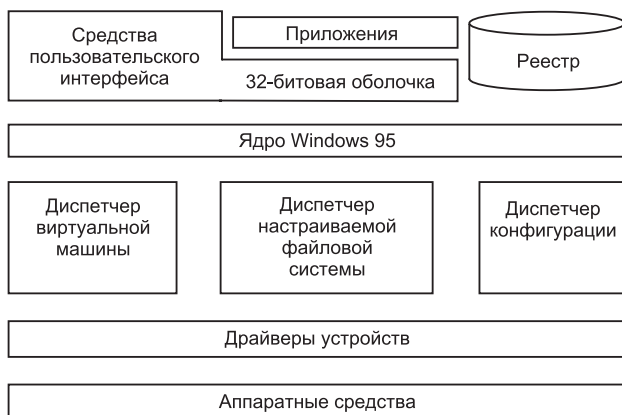


Рис. 2.16. Основные компоненты Windows 95/98

Реестр. Важную роль в управлении ресурсами играет реестр. Реестр содержит всю системную информацию о приложениях,

пользователях и аппаратной части ЭВМ, на основании которой и происходит распределение ресурсов.

Реестр — это иерархическая база данных, в которой централизованно хранится вся информация об аппаратных средствах и о конкретных приложениях Windows 95/98 и о настройках пользователя интерфейсной части ОС.

Драйверы устройств. В Windows 95/98 применяется архитектура «универсальный драйвер — мини-драйвер».

Универсальный драйвер содержит основную часть кода, необходимого для общения целого класса устройств (например, для принтеров или модемов) с соответствующими компонентами операционной системы (скажем, с подсистемами печати или связи).

Мини-драйвер содержит небольшую часть кода, который обеспечивает работу конкретного устройства, принадлежащего данному классу.

Диспетчер конфигурации. Он включен в архитектуру Windows 95/98 для поддержки функциональных возможностей технологии Plug-&-Play.

Диспетчер виртуальной машины. VMM (Virtual Machine Manager) выделяет ресурсы каждому приложению и системному процессу, выполняемому на компьютере.

Виртуальная машина представляет собой некую среду в памяти, которая кажется приложению отдельным компьютером с теми же ресурсами, что и у физического компьютера.

Настраиваемые файловые системы. Файловая система Windows 95/98 характеризуется многоуровневой архитектурой, поддерживающей несколько файловых систем (ФС на основе FAT, файловая система CD ROM, файловые системы сторонних разработчиков).

Особенностью файловой системы Windows 95/98 является поддержка длинных имен файлов (каталогов). В именах файлов можно использовать до 255 символов, включая пробелы и знаки препинания. Запрещенными в именах являются только следующие знаки:

- \ — обратный слэш (используется для обозначения пути);
- / — прямой слэш (используется для подстановки ключей);
- > — знак «больше» (используется для указания направления вывода);
- < — знак «меньше» (используется для указания направления вывода);

: — двоеточие (используется для обозначения имени диска);
? — вопросительный знак (используется в масках поиска);
* — знак «звездочка» (используется в масках поиска);
" — кавычки (в них заключаются полные имена файлов и папок с длинными именами).

Другая особенность — это динамическое кэширование, поддерживаемое файловой системой CD ROM. Это обеспечивает оптимальный баланс между памятью, необходимой приложению, и памятью, выделяемой под дисковый кэш.

Интерфейс Windows 95/98

Интерфейс — важная компонента операционной системы (как, впрочем, и любой программы), предопределяющая эффективность решения поставленных задач и способы работы в программе. В понятие интерфейса входят следующие компоненты:

- внешний вид — как выглядит на экране оболочка программы;
- набор команд, используемый программой;
- способ подачи команд и реакция программ на них.

Экран Windows 95/98. Сразу после установки Windows 95/98 вы видите экран, на котором находится несколько графических объектов. Основные элементы нового интерфейса: Рабочий стол и Линейка задач (рис. 2.17, 1, 9).

Рабочий стол занимает все пространство экрана и вполне соответствует своему названию. Здесь могут располагаться документы, файлы, ярлыки и прочие объекты для удобной работы с ними. Первоначально на Рабочем столе помещены значки Мой компьютер, Сетевое окружение, Корзина и т. п.

Линейка задач — один из основных элементов пользовательского интерфейса Windows 95/98. Задуманная как инструмент для запуска приложений и переключения между ними, Линейка Задач в дальнейшем приобрела гораздо больше функций. Основные элементы линейки — это кнопка Пуск для вызова Главного меню Windows 95/98 и кнопки приложений для переключения между ними.

Во время установки Windows 95/98 запрашивает пользователя, какие именно программы будут использоваться чаще всего. Эти программы помещаются в меню Программы кнопки Пуск и могут запускаться непосредственно оттуда. По желанию пользо-

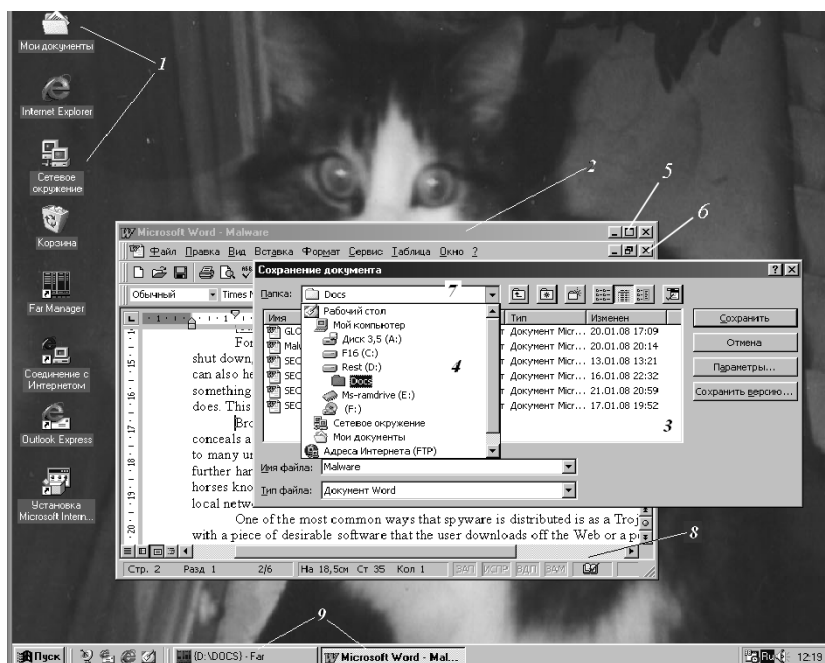


Рис. 2.17. Интерфейс Windows 95/98/2000 с основными элементами:

1 — рабочий стол с ярлыками приложений и файлов; 2 — рабочее окно приложения; 3 — список; 4, 7 — раскрывающийся список; 5 — кнопка развертывания окна; 6 — кнопка закрытия приложения; 8 — линия прокрутки окна; 9 — панель задач (линейка приложений)

вателя список наиболее часто используемых программ легко изменяется в любое время из меню Линейки задач кнопки Пуск. При установке Windows 95/98 поверх предыдущих версий Windows все программные группы автоматически преобразуются в соответствующие папки, содержимое которых доступно с помощью той же кнопки Пуск.

Переключение между задачами. В Windows 95/98 введен механизм (реализуемый с помощью Линейки Задач), позволяющий переключаться между различными активными задачами. При открытии любого окна на Линейке задач автоматически возникает кнопка приложения, находящегося в этом окне. Для перехода к другой задаче следует нажать соответствующую кнопку на Линейке задач.

Размер кнопок на Линейке задач автоматически изменяется в зависимости от количества запущенных задач. Если кнопки,

по мнению пользователя, становятся очень мелкими, он может настроить Линейку Задач по своему усмотрению. Опции конфигурирования Линейки задач позволяют:

- поместить Линейку Задач в любое место по периметру экрана;
- изменить размер Линейки задач, «перетащив мышкой» его внутренний край;
- спрятать Линейку задач с тем, чтобы она появлялась на экране только в том случае, когда курсор мыши попадает на край экрана.

Следует отметить также специально разработанный видеоряд, включающийся при минимизации задачи и помещении ее в виде кнопки на Линейку задач и при обратном процессе восстановления задачи.

Работа с окнами. Основным элементом Windows является окно. Все программы, запущенные из-под Windows, работают в окне. Окна по правилам работы с ними разделяются на рабочие и диалоговые.

Рабочее окно — это, с одной стороны, прямоугольная часть экрана, имеющая стандартные элементы управления, а с другой — представитель программы (именно в окне происходит весь обмен информацией между пользователем и программой).

Все рабочие окна имеют стандартные элементы (рис. 2.17, 2):

- пиктограмму системного меню;
- название окна;
- кнопки: «свернуть» — , «развернуть — восстановить» —  — , «закрыть» — .
- строку меню;
- панель инструментов (может быть включена или выключена);
- рабочее поле окна;
- рамку;
- строку состояния.

Эти управляющие элементы предназначены для реализации *свойств рабочего окна*: изменяемые в размерах; перемещаемые; перекрывающиеся.

Изменяемый размер — окно, оно может быть:

- развернуто на весь экран;
- свернуто до значка;
- получено произвольного размера.

Изменить размер окна можно следующими способами:

- воспользоваться командами системного меню Развернуть, Свернуть, Восстановить;
- воспользоваться кнопками Свернуть, Развернуть-восстановить, которые дублируют команды из системного меню. При этом кнопка Свернуть свертывает окно до кнопки на панели задач. Кнопка Развернуть разворачивает окно на весь экран. После выполнения этой операции кнопка превращается в Восстановить —  (восстанавливает тот размер окна, который был у него до операций Свернуть или Развернуть). Тот же результат можно получить, если произвести двойной щелчок мышью по названию окна.

В режиме произвольного размера очень важным элементом является рамка окна. Точно указав на нее мышью, можно получить новый вид указателя мыши:  или , и если в этот момент нажать левую кнопку мыши и перемещать ее, то можно изменить размер окна.

Перемещаемые — рабочие окна (в режиме произвольного размера) можно произвольно перемещать по экрану. С помощью мыши это делается так: указать на заголовок, зажать левую кнопку мыши, переместить в нужное место и отпустить кнопку. С помощью команды «переместить» из системного меню: подать команду и затем клавишами управления курсором переместить окно.

Перекрывающиеся — активное окно перекрывает собой все другие, развернутые на экране, окна (см. рис. 2.17). Для переключения между окнами можно:

- щелкнуть мышью по любому элементу нужного окна;
- воспользоваться кнопками на панели задач, а для перехода в нужное окно — нажать соответствующую кнопку на панели;
- воспользоваться комбинацией клавиш <Alt+Tab>. Нажав и удерживая клавишу <Alt>, следует нажимать клавишу <Tab>. В центре экрана появляется окно, в котором при каждом нажатии клавиши <Tab> будут переключаться значки всех открытых окон. Если отпустить клавишу <Alt>, активизируется (выйдет на передний план) указанное окно.

Диалоговые окна. Диалоговое окно — это окно, предназначенное для ввода пользователем различной управляющей информации. Диалоговые окна выводятся программами для диалога с пользователем (см. рис. 2.17, 3).

Свойства таких окон следующие:

- это перемещаемые, но неизменяемые (как правило) по размерам окна;
- как правило, это окна-поплавки, т. е. они всегда находятся на переднем плане и до тех пор, пока пользователь не закончит работу с таким окном, невозможно вернуться назад к работе с программой, выведшей это окно;
- такое окно состоит из набора стандартных управляющих элементов (виджетов).

Работа с файлами. В Windows 95/98 для просмотра «содержимого» компьютера используются новые механизмы. Например, существует иконка *Мой компьютер*, которая запускает приложение для работы с файлами, ориентированное на неопытных пользователей. Правда, простота работы с этой программой совсем не означает, что ее возможности ограничены. Так, файлы и «папки» очень логично ведут себя при операциях Drag&Drop и, кроме того, к файлам и «папкам», как и к традиционным объектам — кускам текста, картинкам и т. д. — могут применяться команды *Вырезать*, *Копировать* и *Вставить*.

Drag&Drop — «Перетаски и оставь». Это технология работы с объектами Windows с помощью мыши. Выделив объект (например, файл или папку) и зажав левую кнопку мыши, можно перемещать их на другое место. При этом действия Drag&Drop *по умолчанию* приведут к следующим результатам:

- на одном и том же диске произойдет перемещение объекта;
- перемещение с диска на диск — копирование;
- если объектом является исполняемый файл, произойдет создание ярлыка-ссылки на этот файл.

Однако, этими процессами можно управлять, используя одновременно и управляющие клавиши:

- <Ctrl> — всегда будет производиться копирование;
- <Shift> — всегда будет производиться перемещение.

Если выполнять Drag&Drop правой кнопкой мыши, то по окончании операции выводится контекстное меню, которое позволит выбрать тип производимой операции (*Копировать*, *Переместить*, *Создать ярлык*)

Проводник. Новинкой в Windows 95/98 являлась возможность переименования файлов «на месте» простым щелчком мыши на имени файла с последующим введением нового имени. При этом скрытые расширения остаются неизменными. Файлы могут

переименовываться в диалоговых окнах приложений, таких, как Открыть файл или Сохранить файл.

Столь же просто и удобно производится работа с *сетевыми ресурсами* — папками и принтерами, расположенными на других ПК (если они выделены в совместное использование на последних). Для этого можно использовать приложение Проводник или Сетевое окружение. Другие ПК, которые в настоящий момент работают в локальной сети, просматриваются точно так же, как и дисковые устройства на собственном ПК. При желании может быть подключен любой из доступных ресурсов на другом ПК в качестве сетевого диска.

Предоставление сетевым пользователям доступа к локальной «папке»:

- в Проводнике выбрать одну из «папок» щелчком правой кнопки мыши;
- выбрать пункт Доступ;
- выбрать закладку Доступ и ответить на вопросы в появившемся диалоговом окне.

Подключение стороннего сетевого ресурса в качестве собственного сетевого диска:

- в Проводнике или в окне Сетевое окружение открыть соответствующий ПК;
- выделить ресурс (ресурсы выглядят точно так же, как и папки на локальном ПК);
- щелкнуть на нем правой кнопкой мыши и выбрать команду Подключить сетевой диск...;
- при желании изменить имя диска и установить флажок Автоматически подключать при входе в систему и нажать ОК.

Ярлыки — средство для повышения эффективности работы, особенно полезное в сетевой среде. Пользователь может создать ярлык на любой объект Windows 95/98 (файл, программу, диск, утилиту Панель управления, сетевую папку) и поместить его в любом месте интерфейса или внутри документа. При активизации указателя открывается объект, на который этот указатель ссылается. Например, пользователь может создать ярлык на папку Моя сеть и разместить его на экране Windows 95/98. После этого при активизации ярлыка открывается сетевая папка, находящаяся где-то на сетевом сервере.

Ярлыки выглядят точно так же, как и обычные иконки, за исключением небольшой стрелочки в нижнем левом углу. При создании ярлыка указателя сам объект, на который он ссылался,

остается неизменным. Для создания указателя нужно выбрать объект и выполнить команду Создать ярлык из меню Файл или из контекстного меню, появляющегося при нажатии правой кнопки мыши. Windows 95/98 следит за переименованием и перемещением файлов, оставляя ярлык действительным даже при изменении характеристик объекта, на которые он ссылается (например, при перемещении файла). Ниже приводится лишь несколько вариантов возможного использования ярлыков.

В Windows 95/98 иконки, появляющиеся в меню Пуск\Программы, содержатся в виде ярлыков и в папке Программы. Таким образом, пользователь может сгруппировать ссылки на все свои любимые программы в одном месте независимо от того, где эти программы установлены на самом деле. Когда ярлык устанавливается или удаляется в папке Программы, аналогичный пункт вносится в меню Пуск\Программы или удаляется из него.

Помещение наиболее часто используемых ярлыков прямо на рабочее поле экрана особенно популярно среди пользователей. Таким способом они получают быстрый доступ к наиболее часто используемым ресурсам компьютера; особенно упрощается при этом работа с сетью. Ярлыки, вставляемые в приложения, помогают улучшить организацию данных. Так, например, если пользователь вставит в e-mail-сообщение ярлык на большой файл, находящийся где-либо в сети, то получатель может открыть этот файл в любой момент двойным щелчком мыши на ярлыке. Это эффективнее, чем включать в послание сам файл, поскольку указатель занимает намного меньше места.

Смена пиктограммы ярлыка. Все папки имеют одинаковый вид и различаются только подписями, а ярлыки имеют, как правило, оригинальную пиктограмму (иконку, значок). Можно сменить пиктограмму ярлыка, если щелкнуть по нему правой кнопкой мыши и указать левой кнопкой пункт Свойства в появившемся окне. Затем надо указать пункт Ярлык, нажать кнопки Сменить значок, затем Обзор, выбрать на диске файл, содержащий пиктограммы, выбрать подходящую пиктограмму и нажать две кнопки ОК в верхней и нижней частях окна диалога Свойства.

Пиктограммы могут содержать следующие файлы:

- программы, работающие только в Windows (иногда содержат не один, а несколько значков);
- библиотеки пиктограмм:
 - moricons.dll (путь c:\windows\);
 - progman.exe (c:\windows\);

- shell32.dll (c:\windows\system\);
- pifmgr.dll (c:\windows\system\);
- файлы-иконки *.ico. Их можно при необходимости самому создать в графическом редакторе пиктограмм IconEdit из комплекта Norton Desk Top. В этом редакторе можно создавать целые библиотеки пиктограмм (файлы *.nil).

Окна свойств. С любым объектом пользовательского интерфейса ассоциируется контекстно-зависимый набор свойств, который может быть просмотрен и изменен с помощью меню Файл\Свойства или же с помощью контекстного меню, которое появляется при нажатии правой кнопки мыши. Наилучшим образом пояснят работу окон свойств несколько примеров.

Изменение метки тома:

- в Проводнике или Мой компьютер щелчком правой кнопкой мыши выбрать один из жестких дисков;
- выбрать Свойства;
- ввести новое имя в окошке Метка тома и нажать ОК;
- нажать <F5> (команда Обновить показания в окне).

Изменение значка ярлыка:

- щелкнуть правой кнопкой мыши на любом ярлыке и выбрать Свойства;
- выбрать закладку Ярлык (для Windows-приложений) или Программа (для DOS-приложений);
- нажать кнопку Сменить значок;
- выбрать новую иконку и нажать ОК.

Функции правой кнопки мыши. Наряду с окнами свойств, меню, которые вызываются щелчком правой кнопки мыши, также являются контекстно-зависимыми элементами Windows 95/98 (поскольку при желании пользователи легко могут поменять местами кнопки мыши, везде в тексте под правой кнопкой понимается «второстепенная» кнопка). Ввиду контекстной зависимости меню, вызываемых нажатием правой кнопки мыши, лучше всего пояснить работу описываемого механизма на примерах.

Изменение характеристик рабочего поля экрана на Windows 95/98:

- щелкнуть правой кнопкой мыши на свободном месте экрана, выбрать Свойства;
- сделать необходимые изменения и нажать ОК.

Минимизация и реорганизация окон:

- щелкнуть правой кнопкой мыши на свободном месте Личной панели Задач;

- выбрать **Свернуть все** для сворачивания всех окон или **Сверху вниз** для расположения окон последовательно друг под другом.

Создание нового ярлыка:

- щелкнуть правой кнопкой мыши на объекте, на который следует создать ярлык и выбрать **Создать ярлык**.

Меню Start (Главное меню). Вызывается кнопкой **Пуск** (Start) на линейке задач и служит «стартовой площадкой» не только для поставляемых с Windows 95/98 бесплатных приложений, но и для установленных отдельно программ. После появления меню из него можно выбрать элемент, применяя один из двух способов:

- щелкнуть на элементе мышью;
- при работе с клавиатурой найти подчеркнутую букву в имени элемента, например **P** в Programs, и нажать клавишу с этой буквой. Если подчеркнутой буквы нет, с помощью клавиш управления курсором подсветить элемент и нажать клавишу <Enter>.

Меню Start содержит *команды, принадлежности и приложения*.

Команда — это запуск средства, встроенного в систему Windows 95/98.

Принадлежности — это небольшие прикладные программы, которые поставляются с Windows 95/98.

Приложения представляют собой программы, которые поставляются отдельно и должны быть установлены в компьютере. В большинстве случаев процедура installations просто добавляет имя приложения в меню Start.

Меню Start состоит из меню пользователя (рис. 2.18) и неизменяемой части (рис. 2.19—2.22):

Избранное (меню пользователя). В подменю пользователь может расположить значки приложений, документов, папок и файлов, наиболее необходимых пользователю при работе, для быстрого их вызова. Заполняется с помощью **Пуск\Настройка\Панель задач, закладка Настройка меню** или «перетаскиванием» значка нужного объекта на кнопку **Пуск**.

Неизменяемая часть Главного меню включает:

- **Programs (Программы).** Этот пункт Главного меню содержит сложное иерархическое подменю, состоящее из групп программ и самих программ. Предназначено оно для за-

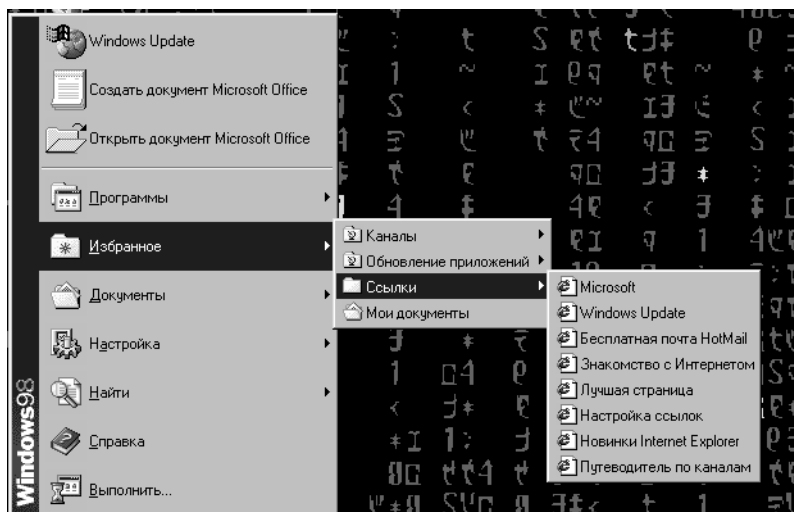


Рис. 2.18. Главное меню (подменю пользователя)

пуска приложений, проинсталлированных в Windows 95/98 (рис. 2.19);

- *Documents (Документы)*. Содержит меню из 15 последних документов, с которыми пользователь работал в любых приложениях. (Под документом подразумевается любой файл, с которым пользователь работал в программе. Им может быть электронная таблица, рисунок, письмо и т. д.) При выборе документа из этой папки Windows запускает соответствующую программу и автоматически загружает документ;
- *Setting (Настройка)*. Содержит несколько программ, с помощью которых можно задать различные параметры Windows 95/98. Включает в себя подменю Панель Управления, Принтеры, Панель Задач.

Панель Управления. Задает параметры системной даты, времени, позволяет производить настройку клавиатуры, мышки, а также установку принтера и модема. Кроме того, позволяет настроить звуковые сигналы на соответствующие события Windows 95/98. Также предоставляет возможность установить конфигурацию сети, идентифицировать компьютер, определить уровень доступа к ресурсам в сети.

Принтеры. Это подменю вызывает один из элементов Панели Управления — модуль для установки и настройки принтеров.

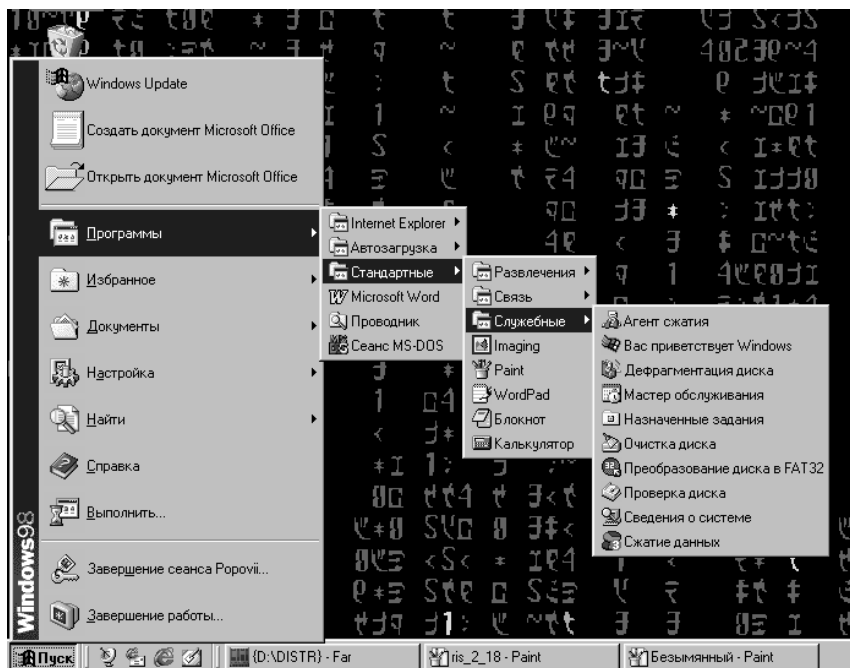


Рис. 2.19. Главное меню. Подменю Программы

Панель Задач включает в себя две закладки:

- Параметры Панели Задач, которая состоит из четырех пунктов:
 - расположить поверх всех окон;
 - автоматически убрать с экрана;
 - мелкие значки в главном меню;
 - отображать часы;
- Настройка меню, включает в себя настройку Главного меню, позволяющую добавлять и удалять пункты меню и управление параметрами панели задач.

Find (Найти). Содержит инструменты, помогающие искать объекты в компьютере. Состоит из подменю Файлы и Папки и подменю Компьютер. Позволяет разыскать нужный файл по имени, размещению, дате изменения, размеру. Все данные выводятся в специальное окно, характерное для Windows 95/98, позволяющее перемещать файлы методом «D&D» и переименовывать (рис. 2.20).

Help (Справка). Вызывает справочную систему Windows 95/98 (рис. 2.21). По сравнению с Windows 3.x, справочная

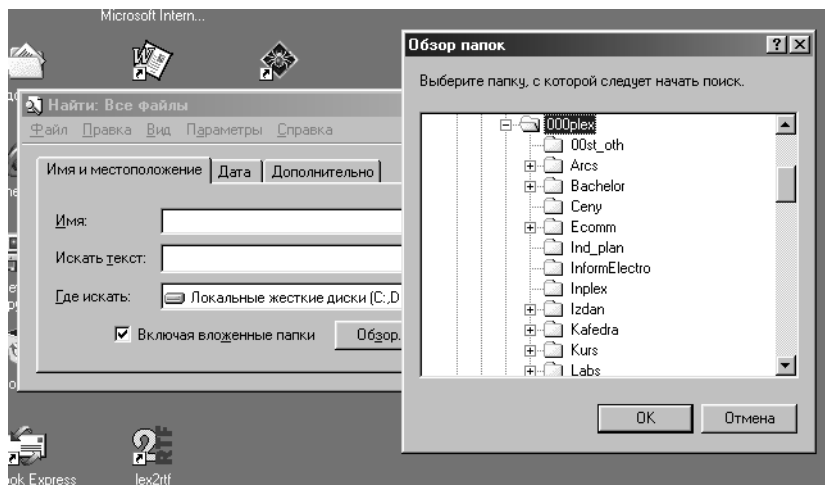


Рис. 2.20. Окно поиска файлов

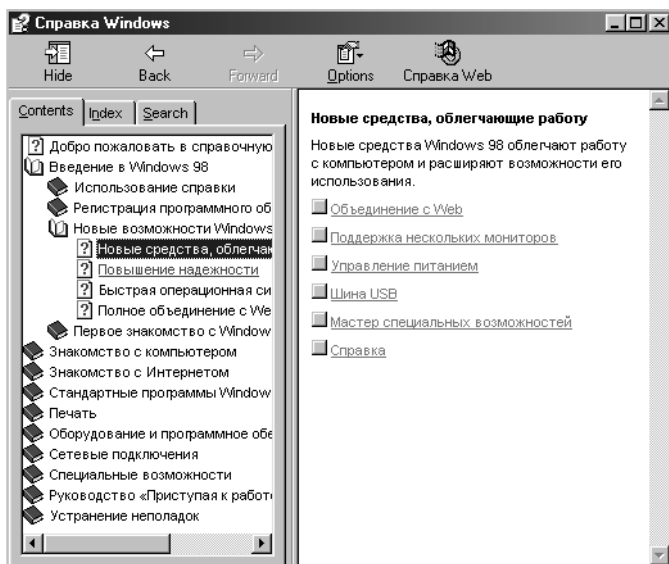


Рис. 2.21. Стартовое меню справочной системы

система Windows 95/98 была полностью переписана. Это вызвано тем, что система помощи в Windows 3.x оказалась сложной для освоения и использования. Она имела три основные функ-

ции — Contents, Search и Glossary, и пользователям не всегда было ясно, какую из них следует использовать в том или ином конкретном случае. Поэтому в Windows 95/98 функция Search не существует, и справочная система содержит только команды Contents и Index.

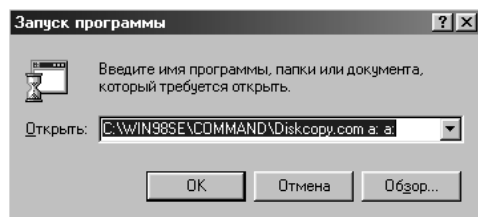


Рис. 2.22. Окно команды запуска приложений

При вызове функции Contents появляется окно, содержание которого похоже на оглавление книги. Темы верхнего уровня сопровождаются иконками с рисунком книги и могут раскрываться для рассмотрения подтем, которые отображаются иконками с изображением листов. Кроме того, все главы укорочены настолько, чтобы они помещались на одном экране и пользователям не приходилось пользоваться прокруткой для рассмотрения объемных сложных тем.

В справочной системе Windows 95/98 появились специальные кнопки-указатели, которые переносят пользователя прямо в то место операционной системы, о котором рассказывается в справке. Так, например, пользователь, читающий в разделе Помощь о том, как переставить часы компьютера, может нажатием одной кнопки попасть в соответствующее место Панели Управления прямо из справочной системы. Кроме того, в Windows 95/98 добавлен новый механизм, уже опробованный в последних версиях Word и Excel компании Microsoft. При нажатии специальной кнопки с изображением вопросительного знака (или выборе соответствующего пункта из контекстного меню, которое вызывается правой кнопкой мыши) курсор принимает форму «?». Если после этого навести его на какой-нибудь объект, справочная система выдаст краткое описание выбранного объекта.

Run (Выполнить). Этой командой пользуются, если нужная программа отсутствует в папках меню Start. После ее выбора нужно ввести дисковый накопитель, где находится программа, ее папку и имя файла запускающего программу (рис. 2.22).

Shut Down (Завершение работы). Эта команда позволяет произвести корректный выход из среды Windows 95/98.

Панель Управления (рис. 2.23) — системная папка, в которой сосредоточены все средства для настройки Windows 95/98.

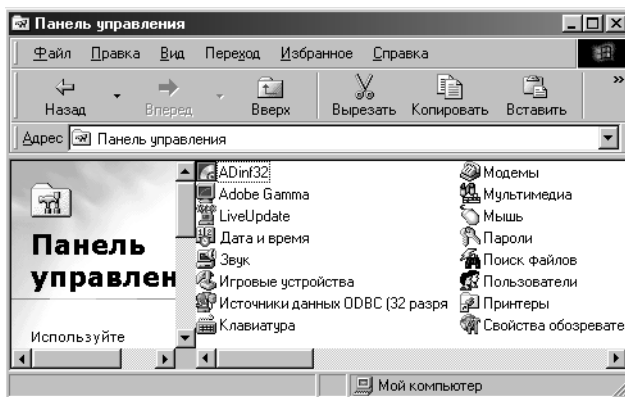


Рис. 2.23. Окно Панели управления

Именно здесь устанавливаются *общие ресурсы Windows*, т. е. такие ресурсы, которые используются всеми приложениями, работающими в данной ОС (операционной системе). Это означает, что данные установки сохраняются самой операционной системой и, в дальнейшем, все приложения используют эти установки для отображения информации. Например, цветовая гамма (цвета заголовка окна, рабочего стола, объемных объектов, рабочих областей приложений и т. д.), национальные установки и многое другое. Кроме того, здесь же производятся и все остальные настройки, такие, как установка нового оборудования, настройка сетевых параметров, добавление и удаление компонентов Windows и пр. Панель управления содержит значки (иконки) для запуска специальных модулей, позволяющих производить перечисленные настройки. Число включенных в нее модулей зависит от полноты инсталляции системы Windows 95/98. Каждая иконка запускает соответствующий модуль, меняющий установки системы. При этом появляется то или иное диалоговое окно.

Ниже перечислены основные пути запуска Панели Управления:

- Пуск\Настройка\Панель Управления;
- Мой компьютер\Панель Управления;

- Проводник — в левом окне выбрать Панель Управления;
- в любой папке (например, Рабочий стол) можно создать ярлык на Панель Управления.

Рассмотрим наиболее важные модули из Панели Управления.

Языки и стандарты. Данный модуль предназначен для установки *общего ресурса Windows* Регион пребывания, который включает в себя: национальные форматы валюты, представления даты и времени, символа, который используется в качестве разделителя между элементами списка, символа-разделителя между дробной и целой частью числа. *Например*, если установить регион пребывания — Россия, то названия месяцев (во всех приложениях, где будет выводиться календарная дата) будут: «январь», «февраль» и т. д., если установить — Украина, то: «січень», «лютий» и т. д. Если установить ресурс разделитель целой и дробной частей числа = «запятая», то запись: «1.5» уже не будет восприниматься приложением (например, Excel) как число, а будет интерпретироваться как текст или календарная дата.

Системный модуль Принтеры. В среде Windows, в отличие от DOS, ни одно приложение не выполняет печать самостоятельно. Приложения только формируют задания на печать и отправляют его специальному приложению — Диспетчеру печати. Windows 95/98 представляет нам диспетчера как установленный принтер конкретной модели. В пакет установки Windows (инсталляционный пакет) входят драйверы, обеспечивающие работу более 800 типов принтеров самого различного типа — от простых 9- и 24-игольчатых матричных принтеров до самых современных цветных струйных и лазерных принтеров с высоким разрешением.

В системе должен быть установлен хотя бы один принтер для того, чтобы приложения Windows могли производить печать документов или показывать предварительный просмотр документа перед печатью. Из любой программы доступна печать как принтерами, установленными на ПК (локально), так и сетевыми принтерами, подключенными к другому ПК или серверу в компьютерной сети.

Один из установленных принтеров назначается *текущим*, т. е. используемым всеми приложениями по умолчанию (если пользователь не указал другой). Этот принтер обеспечивает все приложения Windows информацией о состоянии принтера, о размерах используемой бумаги, ее ориентации и т. д.

Системный модуль Принтеры выглядит, на первый взгляд, как окно обычной папки и позволяет:

- устанавливать драйверы принтеров (т. е. добавлять в систему принтер конкретной модели, например Epson FX-1050);
- настраивать параметры принтеров — команда Свойства из контекстного меню на значке соответствующего принтера.

Шрифты. Windows выгодно отличается от MS DOS возможностью применения множества масштабируемых шрифтов (фонт-тов). Масштабируемость означает возможность изменения размера того или иного шрифта при сохранении достаточно гладкого вида символов. Основным видом фонтов в Windows 95/98 являются хорошо известные пользователям прежних версий Windows фонты True type, сохраняющие свое очертание при изменении размеров в широких пределах.

Системная папка Шрифты предназначена для добавления/удаления шрифтов в среде Windows 95/98 (рис. 2.24). Она запускает специальный модуль для работы с каталогом %WINDIR%/Fonts, с помощью которого можно вывести окно с файлами всех наборов фонтов, установленных в системе, — это наборы, поставляемые с Windows 95/98, наборы, установленные отдельно, и наборы, установленные некоторыми приложениями.

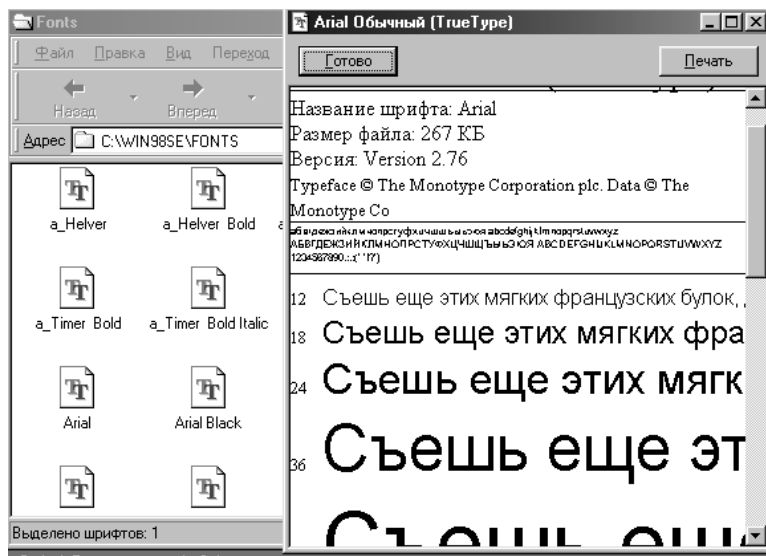


Рис. 2.24. Окно управления шрифтами

Открыв любой файл, можно получить диалоговое окно с подробной характеристикой заданного набора шрифтов.

Помимо информации о наборе фонтов в окне просмотра даны примеры вывода фонтов с различным размером. Опция *Done* позволяет продолжить просмотр, *Print* — распечатать набор принтером. Это полезно для суждения о том, стоит ли применять данный набор для определенных целей, например, печати визитной карточки, письма или приглашения на день рождения.

Установка/удаление программ. Одна из часто используемых функций — установка в Windows новых приложений. Она была возможна и в прежних версиях Windows, но с одним серьезным недостатком — от установленных и затем стертых приложений все же оставались файлы, расходующие память жесткого диска и захламляющие файловую систему.

В Windows 95/98 установка новых программ и переустановка самой системы может происходить разными способами. Например, можно задать новый ярлык и указать, какой программный файл он запускает. Используя команду *Выполнить...* Главного меню и запустив файлы *Setup* или *Install* приложений, можно выполнить их установку или инсталляцию, пополнив таким образом набор приложений, с которыми будет работать Windows 95/98.

Вместе с тем наиболее правильно использовать специальный модуль Панели управления *Установка/удаление программ*, который используется для установки новых программ, а также их корректного удаления из операционной системы.

Диалоговое окно этого модуля на первой закладке отображает список всех установленных в ОС приложений (рис. 2.25). Выбрав нужное приложение в списке и нажав кнопку *Добавить/удалить*, пользователь может запустить окно Мастера, позволяющее удалить данное приложение или изменить состав его компонентов (если это предусмотрено самим приложением, как, например, MS Office).

При нажатии кнопки *Установить* запускается Мастер по установке нового приложения. В этом случае можно проводить установку программ с гибкого диска или CD-ROM. Поскольку инсталляция различных программ имеет свои отличительные особенности, для ее проведения и используется Мастер. Нажимая кнопку *Next* в окне Мастера, можно пройти все шаги по инсталляции той или иной программы. Инсталляция крупных программ занимает много времени — подчас до десятков

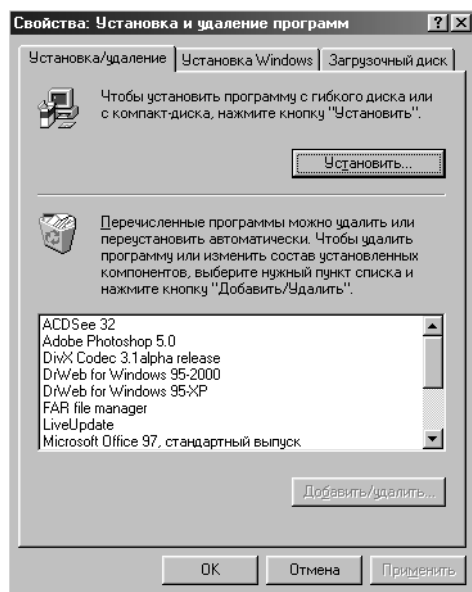


Рис. 2.25. Установка и удаление программ

минут и требует слежения за ней. Тем не менее при применении Мастеров она довольно проста и редко сопровождается сбоями.

Компоненты Windows 95/98 добавляются и удаляются через закладку *Установка Windows*. На ней отображается список тех программ, которые включены в инсталляционный пакет Windows и которые пользователь может по своему усмотрению устанавливать или нет. Установленные компоненты отмечены «галочкой», а неустановленные — нет. Чтобы добавить нужный компонент (например, *Специальные возможности*), следует поставить против него отметку и нажать **ОК**. При выполнении установки Windows потребует указать путь к инсталляционному пакету.

Система. Модуль Система позволяет просматривать и изменять аппаратную конфигурацию системы. Окно свойств системы обеспечивает доступ ко всем параметрам компьютера и отдельных устройств (рис. 2.26).

Закладка Общие — чисто информационная, отображает общие сведения о системе.

Закладка Устройства отображает дерево устройств (физического оборудования), установленных на ПК. В корне дерева находятся названия типов устройств, а ниже — модель кон-

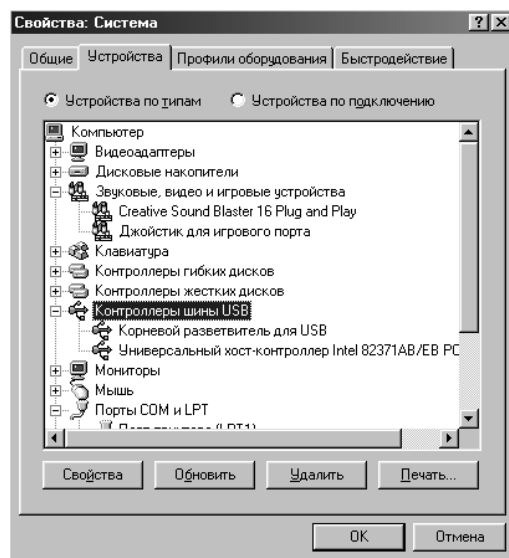


Рис. 2.26. Закладка Устройства

кретного оборудования. Выделив какое-нибудь устройство, можно получить окно с его свойствами, для чего нужно воспользоваться кнопкой **Свойства**. Кнопка **Обновить** обновляет данные дерева устройств, кнопка **Удалить** удаляет выделенное устройство из текущей конфигурации оборудования.

Закладка Конфигурации используется для создания нескольких альтернативных конфигураций оборудования на одном ПК. Использование нескольких конфигураций оборудования позволяет всякий раз при изменении состава оборудования загружать правильный набор драйверов. Это необходимо в первую очередь при работе с переносным компьютером, который может, например, работать как *в доке* (системе стационарного подключения), так и *вне дока*.

Закладка Быстродействие (рис. 2.27) отображает сведения о памяти (ОП), системных ресурсах (имеется в виду сколько свободно памяти = ОП+виртуальная память), файловой системе, а также на этой закладке появляется список причин, замедляющих работу системы.

Если работа системы ничем не замедляется, вместо перечня проблем в окне присутствует сообщение Система настроена на оптимальное быстродействие.

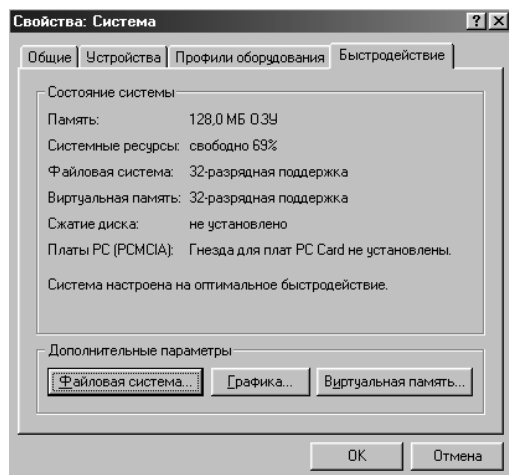


Рис. 2.27. Закладка Быстродействие

Если причины существуют, то выводится список драйверов, работа с которыми замедлена. Чтобы получить более подробные разъяснения по ним, следует выбрать одно из сообщений и нажать кнопку **Сведения** (появляется только при наличии списка причин).

Кнопка *Файловая система* позволяет управлять параметрами эширования при выполнении операций ввода-вывода на накопители (рис. 2.28).

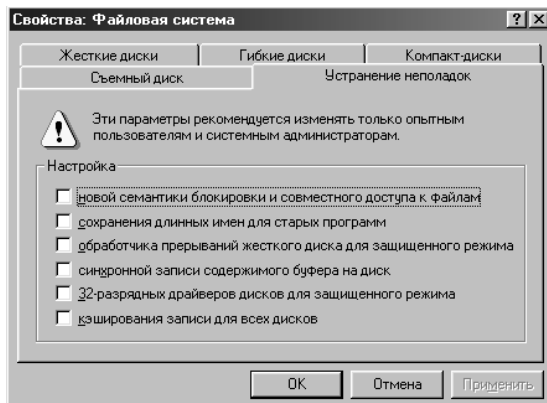


Рис. 2.28. Закладка файловая система — устранение неполадок

Кнопка Графика — настройка параметров ускорения видеоадаптера.

Кнопка Виртуальная память применяется для назначения диска, на котором будет расположен «*файл подкачки*» и задания его размера (используется ОС для виртуального расширения объема используемой памяти). Он имеет большой размер (до двух размеров ОП), и если на системном диске осталось мало дискового пространства, может возникнуть необходимость воспользоваться этой закладкой.

Краткие сведения по архитектуре Windows 95/98

Windows 95/98 представляет собой продукт эволюционного развития систем Windows 3.1x. Хотя она и несет в себе много важных изменений по сравнению с 16-разрядной архитектурой Windows, в ней сохранены некоторые важнейшие свойства ее предшественницы. Результатом стало появление гибридной ОС, способной работать с 16-разрядными прикладными программами Windows, программами, унаследованными от DOS, и старыми драйверами устройств реального режима, и в то же время совместимой с 32-разрядными прикладными программами и 32-разрядными драйверами виртуальных устройств.

Среди наиболее важных усовершенствований, явившихся в Windows 95/98, — заложенная в ней способность работать с 32-разрядными многопоточковыми прикладными программами, защищенные адресные пространства, вытесняющая многозадачность, намного более широкое и эффективное использование драйверов виртуальных устройств и возросшее применение 32-разрядных хипов для хранения структур данных системных ресурсов. Ее наиболее существенный недостаток состоит в относительно слабой защищенности от плохо работающих программ, содержащих ошибки.

Каждая собственная прикладная программа Windows 95/98 «видит» неструктурированное адресное пространство размером 4 Гбайт, в котором размещается она сама плюс системный код и драйверы Windows 95/98 (рис. 2.29). Каждая 32-разрядная прикладная программа выполняется так, как будто она монополюно использует ПК. Код прикладной программы загружается в это адресное пространство между отметками 2 и 4 Гбайт. Хотя 32-разрядные прикладные программы «не видят» друг друга, они могут

В Windows 95 достигнут баланс между производительностью, совместимостью и надежностью. Она обеспечивает быстрое исполнение прикладных программ Win16, Win32 и DOS и совместима с драйверами устройств реального режима. Несмотря на лучшую, чем у Windows 3.1, защищенность, она остается уязвимой с нескольких сторон

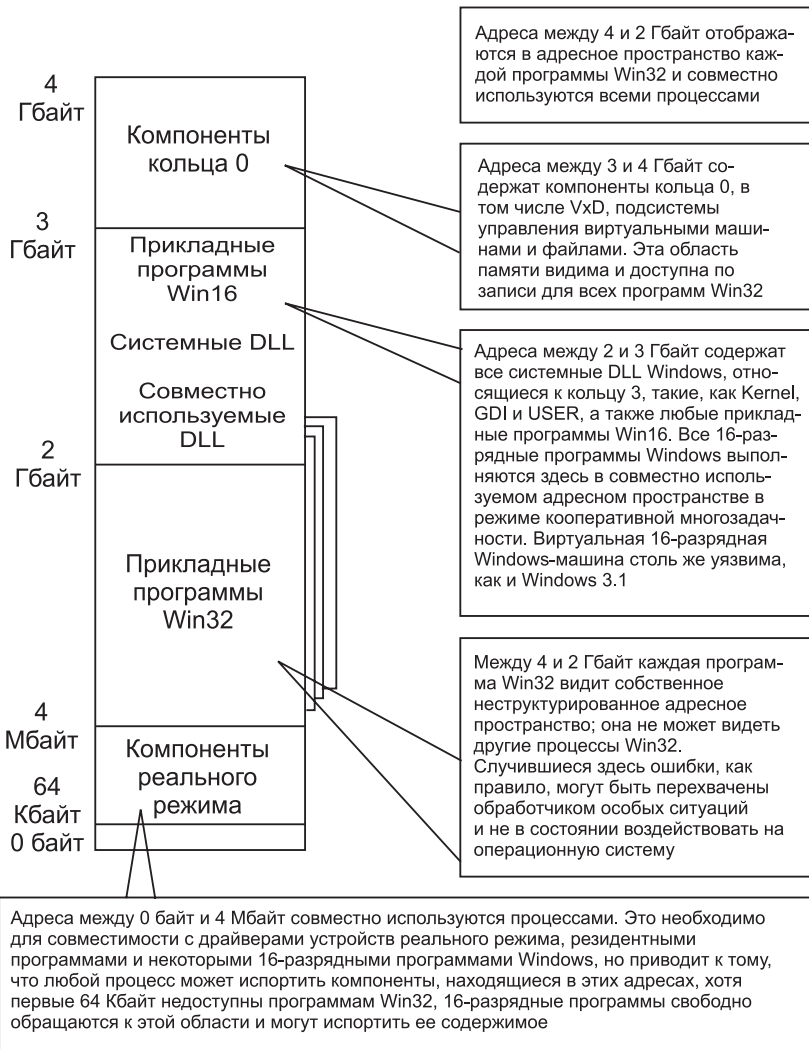


Рис. 2.29. Модель памяти Windows 95/98

обмениваться данными через буфер обмена (Clipboard), механизмы DDE и OLE. Все 32-разрядные прикладные программы выполняются в соответствии с моделью вытесняющей многозадачности, основанной на управлении отдельными потоками. Планировщик потоков, представляющий собой составную часть системы управления виртуальной памятью (VMM), распределяет время среди группы одновременно выполняемых потоков на основе оценки текущего приоритета каждого потока и его готовности к выполнению. Вытесняющее планирование позволяет реализовать намного более плавный и надежный механизм многозадачности, чем кооперативный метод, используемый в Windows 3.1x.

Системный код Windows 95/98 размещается выше границы 2 Гбайт. В пространстве между отметками 2 и 3 Гбайт находятся системные библиотеки DLL кольца 3 и любые DLL, используемые несколькими программами. (В 32-разрядных процессорах фирмы Intel предоставляются четыре уровня аппаратной защиты, поименованные, начиная с кольца 0 до кольца 3, причем кольцо 0 является наиболее привилегированным.) Компоненты кольца 0 в системе Windows 95/98 отображаются в пространство между 3 и 4 Гбайт. Эти важные участки кода с максимальным уровнем привилегий содержат подсистему управления виртуальными машинами (VMM), файловую систему и драйверы VxD.

Область памяти между 2 и 4 Гбайт отображается в адресное пространство каждой 32-разрядной прикладной программы, т. е. оно совместно используется всеми 32-разрядными прикладными программами в вашем ПК. Такая организация позволяет обслуживать вызовы API непосредственно в адресном пространстве прикладной программы и ограничивает размер рабочего множества. Однако за это приходится расплачиваться снижением надежности. Ничто не может помешать программе, содержащей ошибку, произвести запись в адреса, принадлежащие системным DLL, и вызвать крах всей системы.

В области между 2 и 3 Гбайт также находятся все активные 16-разрядные прикладные программы Windows. С целью обеспечения совместимости эти программы выполняются в совместно используемом адресном пространстве, где они могут испортить друг друга так же, как и в Windows 3.1x.

Адреса памяти ниже 4 Мбайт также отображаются в адресное пространство каждой прикладной программы и совместно используются всеми процессами. Благодаря этому становится возможной совместимость с существующими драйверами реального

режима, которым необходим доступ к этим адресам. Это делает еще одну область памяти незащищенной от случайной записи. К самым нижним 64 Кбайт этого адресного пространства 32-разрядные прикладные программы обращаться не могут, что дает возможность перехватывать неверные указатели, но 16-разрядные программы, которые, возможно, содержат ошибки, могут записывать туда данные.

Windows Me



Windows Millennium Edition, также известная как Windows Me, — смешанная 16/32-разрядная операционная система с графическим интерфейсом, выпущенная корпорацией Microsoft 14 сентября 2000 г. Была названа так в честь нового III тысячелетия.

От своих предшественниц — Windows 98 и Windows 95 — новая система отличается относительно небольшими обновлениями, такими как новый Internet Explorer 5.5 и Windows Media Player версии 7. Появился также Movie Maker с базовыми функциями редактирования цифрового видео. Изменился интерфейс системы — в него были добавлены возможности, впервые появившиеся в Windows 2000. Поддержка устройств UPnP (Universal Plug and Play).

Одно из наиболее заметных изменений в Windows Me — в стандартной конфигурации системы отсутствует реальный режим MS DOS, из-за чего нельзя пользоваться программами, требующими его. Однако с помощью специальных утилит эту функцию можно добавить. По сути же загрузка Windows Me ничем не отличается от аналогичного процесса Windows 95 и 98.

В Windows Me впервые появилось много возможностей, характерных для «старших» версий линейки NT: большинство их впервые появилось в Windows 2000, а некоторые — только в Windows XP (табл. 2.5).

Кроме того, в системе были исправлены обнаруженные со времен Windows 98 ошибки, что сделало ее сравнительно стабильной (насколько вообще может быть стабильной система без средств ограничения прав доступа).

Тем самым, по сравнению с Windows 98, система стала функциональнее, причем многие из добавленных функций стали к тому времени повсеместно необходимыми.

Таблица 2.5. Основные нововведения Windows Me

Возможности	Пояснение
Восстановление системы (System restore)	Предоставляет возможность отката системы в предыдущее работоспособное состояние. Технически это выглядит так: по некоторым событиям система создает «точки восстановления», в которые записывает реестр, и затем следит за изменениями в определенном наборе файлов и в случае перезаписи их другими сохраняет в точках восстановления старые версии
Защита системных файлов (Windows File Protection)	Не позволяет перезаписывать довольно значительное множество системных и других файлов никогда, кроме случаев установки обновлений. Призвана решить проблему DLL hell (тупиковая ситуация, связанная с управлением динамическими библиотеками DLL в операционной системе Microsoft Windows)
Автоматическое обновление ОС	Автоматическое обновление с Web-узла Windows Update
Автоматическое распознавание устройств	Распознает съемные устройства USB, Firewire и т. д. и позволяет их безопасно извлекать (в Windows 98 это надо было делать через Device manager или с помощью сторонних утилит)
Справочная система в стиле Windows XP	Локальная справка и поиск на сайте Microsoft
Поддержка нового оборудования	Добавлены драйверы для новых (со времен выхода Windows 98) устройств — пятикнопочная мышь, Web-клавиатура, широкополосные USB-модемы и т. п.
Новые возможности по настройке сети	Поддержка новых технологий подключения: ADSL (Asymmetric Digital Subscriber Loop) и кабельные модемы; общий доступ к подключению Internet
Мастер домашней сети	Вспомогательная программа для простой настройки сети с типовой конфигурацией

Ее невысокая популярность связана скорее с тем, что, во-первых, к моменту своего выхода она уже морально устарела и, во-вторых, не содержала каких-либо кардинальных улучшений по сравнению с Windows 98. Пользователи, которым были нужны новые технологии, как правило, обладали достаточно мощными компьютерами и охотнее переходили на системы линейки NT, имевшие целый ряд существенных достоинств. Обладатели же более старых машин оставались на Windows 98, поскольку им ни поддержка новых технологий, ни тем более косметические улучшения были, как правило, не нужны, и переход на другую ОС не имел особого смысла.

2.5. Операционные системы Windows NT/2000/XP/VISTA/W7

NT или New Technology была создана группой разработчиков под руководством Дэйва Катлера, ранее работавшего в DEC над проектом VMS. Существует легенда, что Дэйв Катлер пришел в Microsoft в 1988 г. специально для работы над проектом NT и поэтому NT, в отличие от остальных ОС Microsoft, в некотором смысле проект одного человека, наиболее законченная ОС.

Windows NT



Windows NT (New Technology) является 32-разрядной операционной системой с приоритетной многозадачностью. В качестве фундаментальных компонент в состав операционной системы входят средства обеспечения безопасности и развитый сетевой сервис. Windows NT также обеспечивает совместимость со многими другими операционными и файловыми системами, а также с сетями. Windows NT способна функционировать как на компьютерах, оснащенных CISC-процессорами со сложной системой команд (complex instruction set computing), так и на компьютерах с RISC-процессорами, имеющими сокращенный набор инструкций (reduced instruction set computing). Операционная система Windows NT также поддерживает высокопроизводительные системы с мультипроцессорной конфигурацией.

Знакомым в Windows NT является только внешний облик. За графическим пользовательским интерфейсом скрываются более мощные возможности.

Задачи, поставленные при создании Windows NT. Система Windows NT не является дальнейшим развитием ранее существовавших продуктов. Ее архитектура создавалась заново с учетом предъявляемых к современной операционной системе требований. Особенности системы, разработанной на основе этих требований, следующие:

- стремясь обеспечить *совместимость* новой операционной системы, разработчики Windows NT сохранили привычный интерфейс Windows и реализовали поддержку существующих файловых систем (таких, как FAT) и различных приложений

(написанных для MS DOS, OS/2 1.x, Windows 3.x и POSIX). Разработчики также включили в состав Windows NT средства работы с различными сетевыми средствами;

- достигнута *переносимость* (portability) системы, которая может теперь работать как на CISC-, так и на RISC-процессорах. К CISC относятся Intel-совместимые процессоры 80386 и выше. RISC представлены системами с процессорами MIPS R4000, Digital Alpha AXP и Pentium серии P54 и выше;
- *масштабируемость* (scalability) означает, что Windows NT не привязана к однопроцессорной архитектуре компьютеров, а способна полностью использовать возможности, предоставляемые симметричными мультипроцессорными системами. В настоящее время Windows NT может функционировать на компьютерах с числом процессоров от 1 до 32. Кроме того, в случае усложнения стоящих перед пользователями задач и расширения предъявляемых к компьютерной среде требований Windows NT позволяет легко добавлять более мощные и производительные серверы и рабочие станции к корпоративной сети. Дополнительные преимущества дает использование единой среды разработки как для серверов, так и для рабочих станций;
- Windows NT имеет однородную *систему безопасности* (security), удовлетворяющую спецификациям правительства США и соответствующую стандарту безопасности B2. В корпоративной среде критическим приложениям обеспечивается полностью изолированное окружение;
- *распределенная обработка* (distributed processing) означает, что Windows NT имеет встроенные в систему сетевые возможности. Windows NT также позволяет обеспечить связь с различными типами хост-компьютеров благодаря поддержке разнообразных транспортных протоколов и использованию средств «клиент — сервер» высокого уровня, включая именованные каналы, вызовы удаленных процедур (RPC — remote procedure call) и Windows-сокеты;
- *надежность и отказоустойчивость* (reliability and robustness) обеспечиваются архитектурными особенностями, которые защищают прикладные программы от повреждения друг другом и операционной системой. Windows NT использует отказоустойчивую структурированную обработку особых ситуаций на всех архитектурных уровнях, которая включает восстанавливаемую файловую систему NTFS и обеспечива-

ет защиту с помощью встроенной системы безопасности и усовершенствованных методов управления памятью;

- возможности *локализации* (allocation) представляют средства для работы во многих странах мира на национальных языках, что достигается применением стандарта Unicode (разработан международной организацией по стандартизации — ISO);
- благодаря модульному построению системы обеспечивается *расширяемость* (insibility) Windows NT, что позволяет гибко осуществлять добавление новых модулей на различных уровнях операционной системы.

Интерфейс Windows NT. При входе в систему пользователь видит образы, практически совпадающие с привычной картиной для Windows 95/98, — рабочий стол, главное меню, линейку приложений (именуется здесь Панель задач).

На рис. 2.30—2.33 приведены типичные экраны и меню, которые наблюдает пользователь Windows NT:

- меню программ, запускаемое из Главного меню;
- экран помощи, также включаемой из Главного меню;
- режим поиска файлов;
- окно настройки линейки приложений (Панели задач) из рубрики Настройки Главного меню.

Внешне интерфейс Windows NT 4.0 аналогичен Windows 95/98. Единственный признак, позволяющий с первого взгляда различить эти две системы, — стартовое меню, где указано, в какой среде вы работаете. В пакет входит ряд прикладных программ — Internet Information Server, Index Server, FrontPage, Internet Explorer, Domain Name System (DNS) Server, Proxy Server и Internet Resource Center, все пакеты Service Pack Plus! и ряд дополнительных утилит, среди которых имеются как новые, например Administrative Wizards или Imager, так и усовершенствованные версии старых программ, например Task Manager.

Administrative Wizard позволяет автоматизировать типичные задачи, возникающие при управлении сетью, а обновленные версии программ Windows NT Diagnostic, Performance Monitor служат для оперативного контроля за состоянием системы. Диалоговое окно Task Manager трансформировалось в мощную программу, которая предоставляет массу полезной информации — от степени загруженности процессора до имен всех активных системных процессоров. При этом пользователь не ос-



Рис. 2.30. Главное меню Windows NT, подменю Программы. Окно установки Свойств экрана

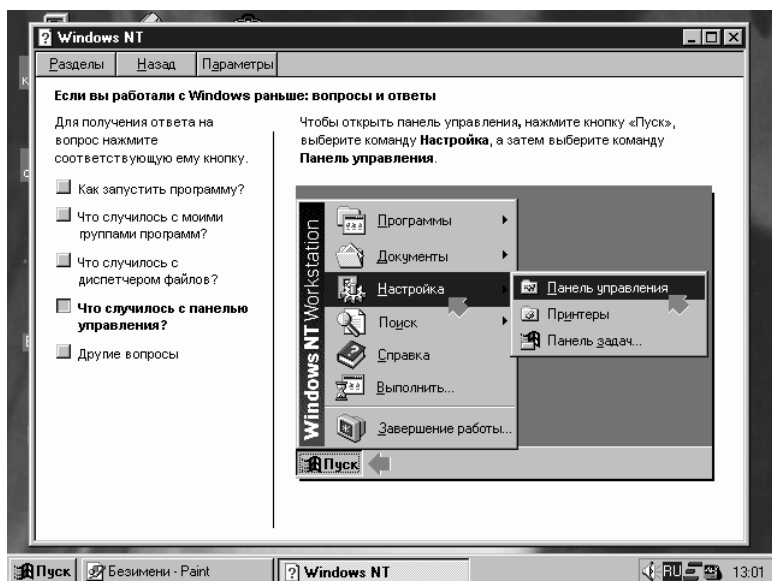


Рис. 2.31. Окно подменю Справка

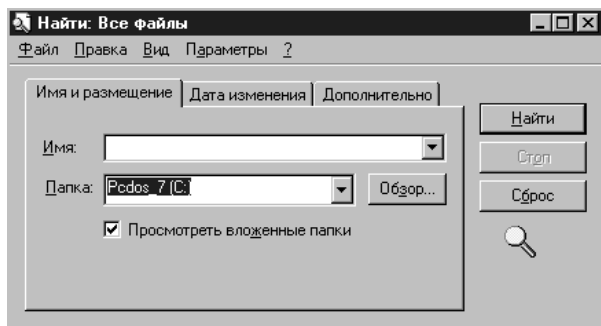


Рис. 2.32. Окно Поиск файлов из Главного меню

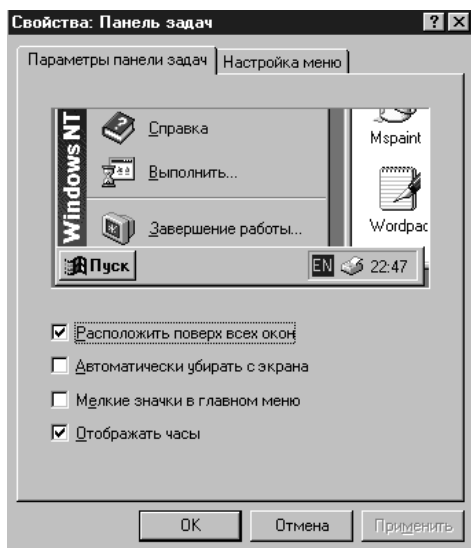


Рис. 2.33. Окно Свойства из пункта Настройка панели задач Главного меню

тается пассивным наблюдателем: при желании с помощью Task Manager возможно, например, завершение любой задачи.

Один из ключевых компонентов Windows NT 4.0 — Internet Information Server. Это гибкое и многофункциональное решение как для подключения к сети Internet, так и для создания собственной частной сети intranet. От пользователя требуется только настроить параметры протокола TCP/IP (при установленном сервисе DHCP IP-адрес присваивается автоматически), запустить IIS и создать одну или несколько собственных Web-страниц. После

этого Web-документы доступны для всех пользователей сети, у которых установлено ПО, обеспечивающее функционирование протокола TCP/IP, и стандартный браузер World Wide Web.

Появились некоторые изменения в подсистеме дистанционного доступа, Remote Access Service (RAS). Теперь имеется возможность использовать защищенные каналы связи, новый протокол Point-To-Point Tunneling Protocol (PPTP), возможность использовать несколько модемов для организации каналов связи с удаленными сетями.

В версии 4.0 были улучшены алгоритмы кэширования сетевых запросов, оптимизированы модули подсистемы разделения ресурсов, изменен механизм генерации прерываний (при переходе к высокоскоростным сетям эта функция неожиданно стала источником проблем для сетевых ОС). Второе изменение, на которое указывает Microsoft, — увеличенная производительность ОС при выполнении графических операций. Разработчики Windows NT перенесли часть кода модулей User и GDI в ядро системы, что позволило ускорить выполнение графических операций на 15—20 %. Однако реальную выгоду от этого улучшения оценить трудно — операции вывода на экран представляют собой лишь малую часть работы, которую выполняют типичные программы для Windows NT. Выводы от более быстрой графики получили преимущественно САПР и ПО для мультимедиа, но даже в этом случае преимущества далеко не очевидны — быстро выполнив запросы на вывод изображения, операционная система, как правило, отдает освободившееся время процессам с более высокими приоритетами.

Она представляла собой основу для нового поколения программных продуктов, ориентированных на работу в сети Internet. Возможность создания инфраструктуры intranet, простота в обращении и хорошая репутация прошлых версий Windows NT в сочетании с усиливающейся тенденцией к созданию однородных сетей сделали ее привлекательной для пользователей из сферы бизнеса.

Архитектурные модули Windows NT

Как показано на рис. 2.34, Windows NT представляет собой модульную операционную систему, которая состоит из отдельных взаимосвязанных относительно простых модулей.

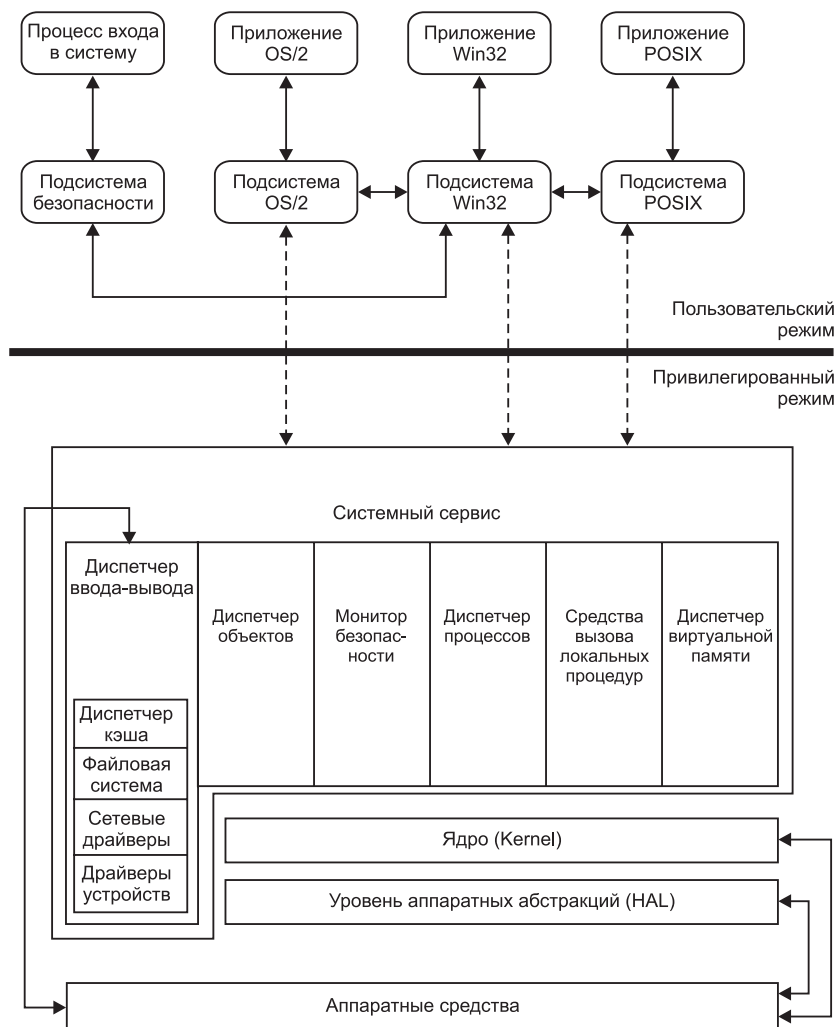


Рис. 2.34. Модульная структура Windows NT

Основными модулями Windows NT являются (перечислены в порядке следования от нижнего уровня архитектуры к верхнему) — уровень аппаратных абстракций HAL (Hardware Abstraction Layer), ядро (Kernel), исполняющая система (Executive), защищенные подсистемы (protected subsystems) и подсистемы среды (environment subsystems).

Уровень аппаратных абстракций обеспечивает независимость остальной части операционной системы от конкретных аппаратных особенностей. Подобный подход позволяет обеспечить легкую переносимость Windows NT с одной аппаратной платформы на другую.

Ядро является основой модульного строения системы и координирует выполнение большинства базовых операций Windows NT. Этот компонент специальным образом оптимизирован по занимаемому объему и эффективности функционирования. Ядро отвечает за планирование выполнения потоков, синхронизацию работы нескольких процессоров, обработку аппаратных прерываний и исключительных ситуаций.

Исполнительная система включает в свой состав набор программных конструкций привилегированного режима (kernel mode), представляющих базовый сервис операционной системы подсистемам среды. Исполняющая система состоит из нескольких компонент, каждая из них предназначена для поддержки определенного системного сервиса. Так, одна из компонент — монитор безопасности (Security Reference Monitor) — функционирует совместно с защищенными подсистемами и обеспечивает реализацию модели безопасности системы.

Подсистемы среды представляют собой защищенные серверы пользовательского режима (user-mode), которые обеспечивают выполнение и поддержку приложений, разработанных для различного операционного окружения (различных операционных систем). Примером подсистем среды могут служить подсистемы Win32 и OS/2.

Уровень аппаратных абстракций (HAL) представляет собой создаваемый производителями аппаратных средств слой программного обеспечения, который скрывает (или абстрагирует) особенности и различия аппаратуры от верхних уровней операционной системы. Таким образом, благодаря обеспечиваемому HAL фильтру, различные аппаратные средства выглядят аналогично с точки зрения операционной системы; снимается необходимость специальной настройки операционной системы под используемое оборудование.

При создании уровня аппаратных абстракций ставилась задача подготовки процедур, которые позволяли бы единственному драйверу конкретного устройства поддерживать функционирование этого устройства для всех платформ. HAL ориентирован на большое число разновидностей аппаратных платформ с одно-

процессорной архитектурой; таким образом для каждого из аппаратных вариантов не требуется отдельной версии операционной системы.

Процедуры HAL называются как средствами операционной системы (включая ядро), так и драйверами устройств. При работе с драйверами устройств уровень аппаратных абстракций обеспечивает поддержку различных технологий ввода-вывода (вместо традиционной ориентации на одну аппаратную реализацию или требующей значительных затрат адаптации под каждую новую аппаратную платформу).

Уровень аппаратных абстракций позволяет также «скрывать» от остальных уровней операционной системы особенности аппаратной реализации симметричных мультипроцессорных систем.

Ядро (Kernel) работает в тесном контакте с уровнем аппаратных абстракций. Этот модуль, в первую очередь, занимается планированием действий процессора. В случае, если компьютер содержит несколько процессоров, ядро синхронизирует их работу с целью достижения максимальной производительности системы.

Ядро осуществляет диспетчеризацию *потоков* (threads — нитей управления, иногда называемых подзадачами, ответвлениями), которые являются основными объектами планирования в системе. Потоки определяются в контексте процесса; процесс включает адресное пространство, набор доступных процессу объектов и совокупность выполняемых в контексте процесса потоков управления. Объектами являются управляемые операционной системой ресурсы.

Ядро производит диспетчеризацию потоков управления таким образом, чтобы максимально загрузить процессоры системы и обеспечить первоочередную обработку потоков с более высоким приоритетом. Всего существует 32 значения приоритетов, которые сгруппированы в два класса: *real-time* и *variable*. Подобный подход позволяет достичь максимальной эффективности операционной системы.

Подкомпоненты исполняющей системы, такие, как диспетчер ввода-вывода и диспетчер процессов, используют ядро для синхронизации действий. Они также взаимодействуют с ядром для более высоких уровней абстракции, называемых *объектами ядра*; некоторые из этих объектов экспортируются внутри пользовательских вызовов интерфейса прикладных программ (API).

Ядро управляет двумя типами объектов.

Объекты диспетчеризации (dispatcher objects) характеризуются сигнальным состоянием (signaled или nonsignaled) и управляют диспетчеризацией и синхронизацией системных операций. Эти объекты включают события, мутанты, мутэксы, семафоры, потоки управления и таймеры (events, mutants, mutexes, semaphores, threads, timers).

Управляющие объекты (control objects) используются для операций управления ядра, но не воздействуют на диспетчеризацию или синхронизацию.

Управляющие объекты включают в себя асинхронные вызовы процедур, прерывания, уведомления и состояния источника питания, процессы и профили (asynchronous procedure calls, interrupts, power notifies, power statuses, processes, profiles).

Исполнительная система (Executive), в состав которой входит ядро и уровень аппаратных абстракций HAL, обеспечивает общий сервис системы, который могут использовать все подсистемы среды. Каждая группа сервиса находится под управлением одной из отдельных составляющих исполняющей системы:

- диспетчера объектов (Object Manager);
- диспетчера виртуальной памяти (Virtual Memory Manager);
- диспетчера процессов (Process Manager);
- средства вызова локальных процедур (Local Procedure Call Facility);
- диспетчера ввода-вывода (I/O Manager);
- мониторов безопасности (Security Reference Monitor).

Монитор безопасности совместно с процессором входа в систему (Logon) и защищенными подсистемами реализует *модель безопасности Windows NT*.

Верхний уровень исполняющей системы называется системным сервисом (System Services). Показанный на рис. 2.30 системный сервис представляет собой интерфейс между подсистемами среды пользовательского режима и привилегированным режимом.

Диспетчер кэша. Архитектура ввода-вывода содержит единственный диспетчер кэша (Cache Manager), который осуществляет кэширование для всей системы ввода-вывода. Кэширование (Caching) — метод, используемый файловой системой для увеличения эффективности. Вместо непосредственной записи и считывания с диска часто используемые файлы временно сохра-

няются в кэш-памяти; таким образом, работа с этими файлами выполняется в памяти. Операции с данными, находящимися в памяти, производятся значительно быстрее операций с данными на диске.

Диспетчер кэша использует модель отображения файла, которая интегрирована с диспетчером виртуальной памяти Windows NT. Диспетчер кэша обеспечивает службу кэширования для всех файловых систем и сетевых компонентов, функционирующих под управлением диспетчера ввода-вывода. В зависимости от объема доступной оперативной памяти диспетчер кэша может динамически увеличивать или уменьшать размер кэш-памяти. Когда процесс открывает файл, который уже находился в кэше, диспетчер кэша просто копирует данные из кэша в виртуальное адресное пространство.

Диспетчер кэша поддерживает службы типа замедленной записи (lazy write) и замедленной фиксации (lazy commit), которые могут значительно увеличить эффективность файловой системы. В процессе замедленной записи изменения регистрируются в кэше файловой структуры, обеспечивающем более быстрый доступ. Позднее, когда загрузка центрального процессора снижена, диспетчер кэша заносит изменения на диск. Замедленная фиксация подобна замедленной записи. Вместо немедленной маркировки транзакции как успешно завершившейся, переданная информация кэшируется и позднее в фоновом режиме записывается в журнал файловой системы.

Драйверы файловой системы. В архитектуре ввода-вывода Windows NT управление драйверами файловой системы осуществляет диспетчер ввода-вывода. Windows NT допускает использование множества файловых систем. Для обеспечения совместимости снизу вверх с операционными системами MS DOS, Windows 3.x и OS/2 Windows NT поддерживает файловые системы FAT и HPFS.

Кроме того, Windows NT также поддерживает NTFS — новую файловую систему, разработанную специально для использования с Windows NT. NTFS обеспечивает ряд возможностей, включая средства восстановления файловой системы, поддержку Unicode, длинных имен файлов и поддержку для POSIX.

Архитектура ввода-вывода Windows NT не только поддерживает традиционные файловые системы, но и обеспечивает функционирование сетевых серверов в качестве драйверов файловой системы. С точки зрения диспетчера ввода-вывода, нет разницы

между работой с файлом, размещенным на удаленном компьютере сети, и работой с файлом на локальном жестком диске. Серверы могут быть загружены и выгружены динамически так же, как и любые другие драйверы; на одном компьютере может одновременно находиться большое число серверов.

Сетевые драйверы. Следующим типом драйверов, присутствующих в качестве компонента в архитектуре ввода-вывода, являются сетевые драйверы. Windows NT включает интегрированные возможности работы с сетями и поддержку для распределенных приложений. Серверы функционируют как драйверы файловой системы и выполняются на уровне интерфейса поставщика или ниже.

Драйверы транспортного протокола общаются с серверами через уровень, называемый интерфейсом транспортного драйвера (TDI — Transport Driver Interface). Windows NT включает следующие транспортные средства:

- протокол управления передачей/межсетевой протокол TCP/IP, который обеспечивает возможность работы с широким диапазоном существующих сетей;
- NBF, потомок расширенного интерфейса пользователя NetBIOS (NetBEUI), который обеспечивает совместимость с существующими локальными вычислительными сетями на базе LAN Manager, LAN Server и MS-Net;
- управление передачей данных (DLC — Data Link Control), которое обеспечивает интерфейс для доступа к мэйнфреймам и подключенным к сети принтерам;
- NWLink, реализация IPX/SPX, обеспечивающая связь с Novell NetWare.

В нижней части сетевой архитектуры находится драйвер платы сетевого адаптера. Windows NT в настоящее время поддерживает драйверы устройств, выполненные в соответствии со спецификацией NDIS (Network Device Interface Specification) версии 3.0. NDIS предоставляет гибкую среду обмена данными между транспортными протоколами и сетевыми адаптерами. NDIS 3.0 позволяет отдельному компьютеру иметь несколько установленных в нем плат сетевых адаптеров. В свою очередь, каждая плата сетевого адаптера может поддерживать несколько транспортных протоколов для доступа к различным типам сетевых станций.

Модель безопасности Windows NT представлена монитором безопасности (Security Reference Monitor), а также

двумя другими компонентами: процессором входа в систему (Logon Process) и безопасными защищенными подсистемами.

В многозадачной операционной системе, каковой является Windows NT, приложения совместно используют ряд ресурсов системы, включая память компьютера, устройства ввода-вывода, файлы и процессор(ы) системы. Windows NT включает набор компонентов безопасности, которые гарантируют, что приложения не смогут обратиться к этим ресурсам без соответствующего разрешения.

Монитор безопасности отвечает за проведение в жизнь политики проверки правильности доступа и контроля, определенной локальной подсистемой безопасности. Монитор безопасности обеспечивает услуги по подтверждению доступа к объектам, проверке привилегий пользователя и генерации сообщений как для привилегированного режима, так и для режима пользователя. Монитор безопасности, подобно другим частям операционной системы, выполняется в привилегированном режиме.

Процесс входа в систему в Windows NT предусматривает обязательный вход в систему безопасности для идентификации пользователя. Каждый пользователь должен иметь бюджет и использовать пароль для обращения к этому бюджету.

Прежде чем пользователь сможет обратиться к любому ресурсу компьютера из Windows NT, он должен войти в систему через процесс входа в систему для того, чтобы подсистема безопасности могла распознать имя пользователя и пароль. Только после успешного установления подлинности монитор безопасности выполняет процедуру проверки правильности доступа для определения права пользователя на обращение к этому объекту.

Защищенность ресурсов — одна из особенностей, предоставляемая моделью безопасности. Задачи не могут обращаться к чужим ресурсам (типа памяти) иначе, чем через применение специальных механизмов совместного использования.

Windows NT также предоставляет средства контроля, которые позволяют администратору фиксировать действия пользователей.

Управление памятью Windows NT. Windows NT Workstation 3.51 по существу представляет собой операционную систему сервера, приспособленную для использования на рабочей станции. Этим обусловлена архитектура, в которой абсолютная защита прикладных программ и данных берет верх над сообще-

Архитектура клиент-сервер Windows NT обеспечивает существенно более надежную защиту, чем Windows 95. Но для получения приемлемой производительности требуется мощный ПК с обширной памятью. Windows NT работает с программами, ориентированными на Win16 и DOS, но драйверы устройств реального режима не могут функционировать в среде Windows NT

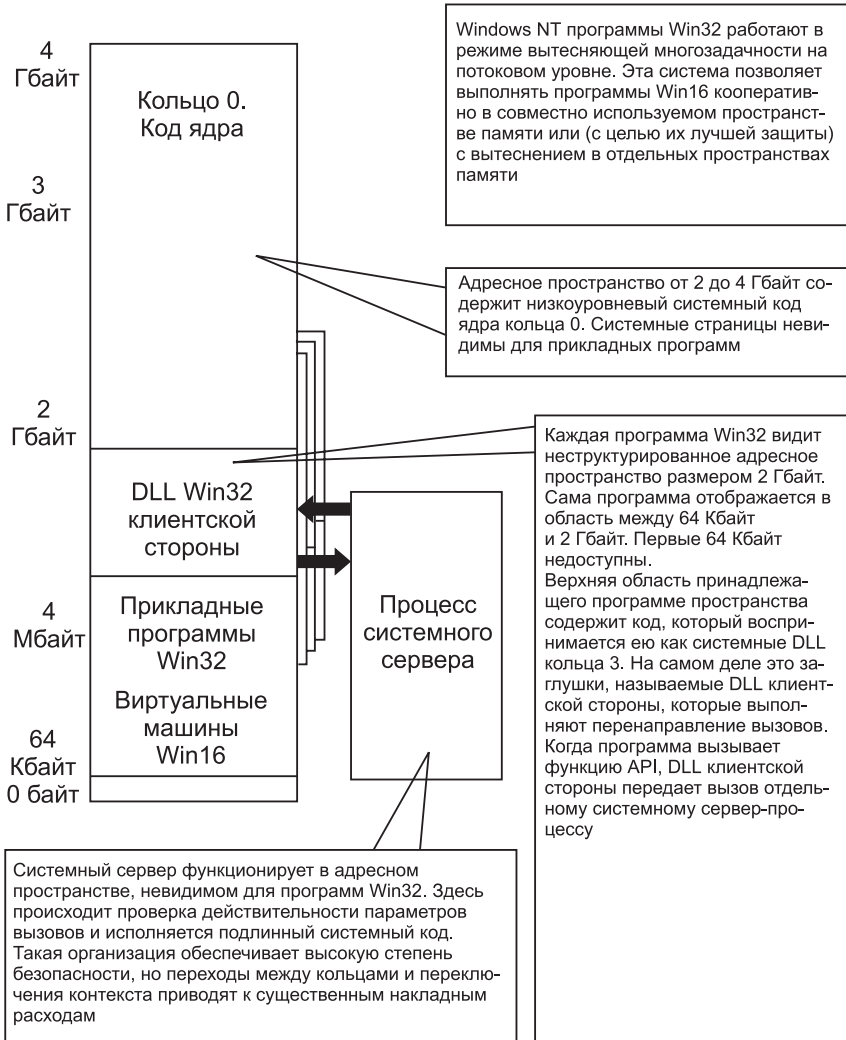


Рис. 2.35. Модель памяти Windows NT

ниями скорости и совместимости. Чрезвычайная надежность Windows NT обеспечивается ценой высоких системных затрат, поэтому для получения приемлемой производительности необходимы быстродействующий ЦП и по меньшей мере 16-Мбайт ОП. В системе Windows NT безопасность нижней памяти достигается за счет отказа от совместимости с драйверами устройств реального режима. В среде Windows NT работают собственные 32-разрядные NT- прикладные программы, а также большинство прикладных программ Windows 95/98. Так же, как и Windows 95/98, система Windows NT позволяет выполнять в своей среде 16-разрядные Windows- и DOS-программы.

Схема распределения памяти Windows NT отличается от распределения памяти системы Windows 95/98 (рис. 2.35). Собственным прикладным программам выделяется 2 Гбайт особого адресного пространства, от границы 64 Кбайт до 2 Гбайт (первые 64 Кбайт полностью недоступны). Прикладные программы изолированы друг от друга, хотя могут общаться через буфер обмена Clipboard, механизмы DDE и OLE.

В верхней части каждого 2-Гбайт блока прикладной программы размещен код, воспринимаемый прикладной программой как системные библиотеки DLL кольца 3. На самом деле это просто заглушки, выполняющие перенаправление вызовов, называемые DLL клиентской стороны (client-side DLLs). При вызове большинства функций API из прикладной программы библиотеки DLL клиентской стороны обращаются к локальным процедурам (Local Process Communication — LPC), которые передают вызов и связанные с ним параметры в совершенно изолированное адресное пространство, где содержится собственно системный код. Этот сервер-процесс (server process) проверяет значение параметров, исполняет запрошенную функцию и пересылает результаты назад в адресное пространство прикладной программы. Хотя сервер-процесс сам по себе остается прикладного уровня, он полностью защищен от вызывающей его программы и изолирован от нее.

Между отметками 2 и 4 Гбайт расположены низкоуровневые системные компоненты Windows NT кольца 0, в том числе ядро, планировщик потоков и диспетчер виртуальной памяти. Системные страницы в этой области наделены привилегиями супервизора, которые задаются физическими схемами кольцевой защиты процессора. Это делает низкоуровневый системный код невидимым и недоступным по записи для программ прикладного

уровня, но приводит к падению производительности во время переходов между кольцами.

Для 16-разрядных прикладных Windows-программ Windows NT реализует сеансы Windows on Windows (WOW). Windows NT дает возможность выполнять 16-разрядные программы Windows индивидуально в собственных пространствах памяти или совместно в разделяемом адресном пространстве. Почти во всех случаях 16- и 32-разрядные прикладные программы Windows могут свободно взаимодействовать, используя OLE (при необходимости через особые процедуры *thunk*) независимо от того, выполняются они в отдельной или общей памяти. Собственные прикладные программы и сеансы WOW выполняются в режиме вытесняющей многозадачности, основанной на управлении отдельными потоками. Множественные 16-разрядные прикладные программы Windows в одном сеансе WOW выполняются в соответствии с кооперативной моделью многозадачности. Windows NT может также выполнять в многозадачном режиме несколько сеансов DOS. Поскольку Windows NT имеет полностью 32-разрядную архитектуру, не существует теоретических ограничений на ресурсы GDI и USER.

Основные отличия Windows 2000



Windows 2000 или W2k — операционная система Microsoft, основанная на технологии Windows NT, что было отражено в первоначальном названии проекта W2k — Windows NT 5.0. Windows 2000 — полностью 32-разрядная ОС с приоритетной многозадачностью и улучшенной реализацией работы с памятью.

Active Directory — средство управления пользователями и сетевыми ресурсами. Оно призвано облегчить работу администраторам больших сетей на базе W2k, и вокруг него строится вся система управления сетью и ее безопасности. Active Directory строится на следующих принципах:

- единая регистрация в сети — благодаря технологии IntelliMirror, можно подойти к любому компьютеру в офисе, ввести свой пароль и перед пользователем появится его рабочий стол, документы и настройки;
- безопасность информации — в службу Active Directory встроены средства идентификации пользователя. Для каж-

дого объекта в сети можно централизованно выставить права доступа, в зависимости от групп и конкретных пользователей. Благодаря системе безопасности Kerberos можно осуществлять защищенную связь даже по открытым сетям, таким как Internet. При этом данные, передаваемые по сети, шифруются, а пароли не передаются и не хранятся на клиентских машинах;

- централизованное управление — при использовании службы Active Directory у администратора отпадает необходимость вручную конфигурировать каждую машину, если, к примеру, необходимо поменять права доступа к какому-либо одному объекту или установить новый сетевой принтер. Такие изменения можно производить сразу для всей сети;
- гибкий интерфейс — структуры каталогов меняются быстро и легко. Например, можно создать каталог своей фирмы, выделить в отдельные подкаталоги бухгалтерию, отделы маркетинга, секретариат, и представить все это в виде древовидной структуры;
- интеграция с DNS — Active Directory в локальной сети использует те же имена ресурсов, что и в Internet, что способствует более тесному взаимодействию локальной и глобальной сети;
- масштабируемость — несколько доменов Active Directory могут быть объединены вместе под одним управлением;
- простота поиска — в домене Active Directory различные объекты можно находить по самым различным признакам, таким как имя пользователя или компьютера, адрес электронной почты пользователя и т. д.

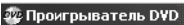
DFS (Distributed File System) — один из инструментов Active Directory. Он позволяет создавать сетевые ресурсы, в которые может входить множество файловых систем на различных машинах. Для пользователя Active Directory это абсолютно прозрачно и не имеет никакого значения, где и на каких машинах физически расположены те файлы, с которыми он работает. Кроме этого, при использовании DFS и Active Directory упрощается управление такими ресурсами.

В табл. 2.6 приводятся основные аксессуары, службы и функции, отличающие W2K от более ранних версий ОС (см. также рис. 2.36).

Таблица 2.6. Программы, службы и функции, встроенные в Windows 2000 Professional

Программа	Функция	Запуск
Адресная книга	Представляет собой место хранения сведений о контактах, позволяющее запросить их из таких программ, как Outlook, Outlook Express, Internet Explorer и NetMeeting. Здесь можно выполнять поиск людей и организаций, создавать группы контактов для рассылки почты и получать электронные визитные карточки	Пуск\Программы\Стандартные, выбрать  Адресная книга
Оснастка Анализ и настройка безопасности	Проверяет и анализирует настройки безопасности системы и рекомендует внесение изменений в текущие настройки системы. Области, в которых текущие настройки не отвечают предполагаемому уровню безопасности, выделяются с помощью значков и комментариев. Администраторы могут использовать оснастку Анализ и настройка безопасности для настройки политики безопасности и обнаружения слабых мест, возникающих в безопасности системы	
Программа архивации	Помогает создать копию данных на жестком диске. Если исходные данные будут случайно удалены, заменены или станут недоступны из-за неисправности жесткого диска, можно использовать резервную копию для их восстановления. Можно также использовать для создания диска аварийного восстановления (ERD)	Пуск\Программы\Стандартные\Служебные  Архивация данных
Блокнот	Текстовый редактор, используемый для создания простых документов или веб-страниц. Для создания и изменения файлов, содержащих форматирование, следует использовать редактор WordPad	Пуск\Программы\Стандартные\  Блокнот
Paint	Инструмент рисования, позволяющий создавать черно-белые или цветные рисунки и сохранять их в виде файлов	Пуск\Программы\Стандартные\выбрать  Paint
Групповая политика	Инструмент администратора для определения и управления параметрами работы программ, сетевых ресурсов и операционной системы для пользователей и компьютеров организации. В среде Active Directory групповая политика применяется к пользователям или компьютерам на основе их принадлежности к узлам, доменам или организационным подразделениям	
Программа дефрагментации дисков	Переупорядочивает файлы, программы и неиспользуемое пространство на жестком диске компьютера таким образом, чтобы сделать более быстрым открытие файлов и запуск программ	Пуск\Программы\Стандартные\Служебные и Выбрать нужную программу  Дефрагментация диска

Продолжение табл. 2.6

Программа	Функция	Запуск
Диспетчер синхронизации	Позволяет задать режим синхронизации автономных файлов с файлами, расположенными в сети. Это обеспечивает получение последних данных из локальной сети или Internet с минимальными неудобствами для работы. Диспетчер синхронизации может автоматически синхронизировать данные, доступные в автономном режиме, несколькими способами: при каждом входе в систему, выходе из нее либо в обоих случаях; в заданные интервалы времени при простое компьютера; в запланированное время. Для автономных файлов из разных общих ресурсов допускается использование различных параметров и комбинации этих параметров	
Диспетчер задач Windows	Предоставляет сведения о производительности компьютера, а также о выполняемых компьютером программах и процессах. С помощью диспетчера задач можно завершать выполнение программ или процессов, запускать программы, а также просматривать динамику производительности компьютера	Щелкнуть правой кнопкой пустое место на панели задач и выбрать в контекстном меню 
Диспетчер устройств	Предоставляет сведения об установленном на компьютере оборудовании и его настройках, а также о взаимодействии этого оборудования с программами компьютера. Кроме того, можно использовать диспетчер устройств для проверки состояния оборудования и обновления драйверов установленных на компьютере устройств	Пуск\Настройка\Панель управления\Система\Оборудование и нажать 
Окно просмотра событий (рис. 2.36, а)	Ведутся журналы программных, системных событий, а также событий безопасности на компьютере. Окно просмотра событий используется для просмотра журналов событий и управления ими, получения сведений о неполадках аппаратного и программного обеспечения, а также для наблюдения за событиями безопасности Windows 2000	Пуск\Настройка\Панель управления\Администрирование\  Просмотр событий
Проигрыватель DVD (рис. 2.36, б)	При вставке диска DVD в дисковод проигрыватель DVD начинает воспроизведение диска автоматически. Для просмотра фильмов в формате DVD необходимо установить на компьютере декодер DVD и дисковод DVD	Пуск\Программы\Стандартные\Развлечения 
Калькулятор	Имеет два режима работы: обычный, предназначенный для простейших вычислений, и инженерный, который обеспечивает доступ ко многим математическим (в том числе и статистическим) функциям	Пуск\Программы\Стандартные и выбрать 

Продолжение табл. 2.6

Программа	Функция	Запуск
Программа Лазерный проигрыватель	Предназначена для воспроизведения звуковых компакт-дисков с подключенного к компьютеру дисковода компакт-дисков. При установке компакт-диска в дисковод Windows 2000 начинает воспроизведение компакт-диска автоматически	Пуск\Программы\Стандартные\Развлечения И выбрать  Лазерный проигрыватель
Оснастка Локальные пользователи и группы	Управляет пользователями и группами пользователей выбранного компьютера. Допускается создание новых пользователей и групп, добавление пользователей в группы, удаление пользователей из групп, отключение учетных записей пользователей и групп, а также сброс паролей	Управление компьютером\  Локальные пользователи
Программа Управление дисками (рис. 2.36, в)	Графическое средство для управления дисками и томами. Поддерживает разделы, логические диски, новые динамические тома и удаленное управление дисками	Пуск\Настройка\Панель управления\Администрирование\Управление компьютером\Запоминающие устройства\Управление дисками  Запоминающие устройства  Управление дисками
Компонент Общие папки	Содержит три списка: Ресурсы — список всех общих файлов и папок на локальном компьютере; Сеансы — список всех пользователей, подключенных к локальному компьютеру; Открытые файлы — список всех файлов локального компьютера, открытых другими пользователями. Компонент Общие папки позволяет создавать, просматривать общие файлы и папки, а также задавать разрешения для доступа к ним	Управление компьютером\  Общие папки
Программа Доктор Ватсон (рис. 2.36, г)	Обнаруживает сведения о сбоях системы и программ и записывает их в файл журнала. При возникновении программной ошибки программа Доктор Ватсон запускается автоматически. Она не может предотвратить возникновение ошибок, но сведения, записываемые в файл журнала, могут использоваться техническим персоналом для диагностики	Пуск\Выполнить, ввести drwtsn32  Доктор Ватсон

Продолжение табл. 2.6

Программа	Функция	Запуск
Программа очистки диска	Помогает очистить пространство на жестком диске. Программа очистки диска проверяет диск и выводит перечень временных файлов, файлов кэша Internet, а также ненужных программных файлов, удаление которых не приведет к негативным последствиям. Можно выбрать удаление некоторых или всех этих файлов	Пуск\Программы\Стандартные\Служебные и Выбрать  Очистка диска
Проигрыватель Windows Media	Позволяет воспроизводить видео, аудио и смешанные файлы. Допускается выполнение таких задач, как просмотр обновления новостей Internet в режиме реального времени, воспроизведение фрагментов из фильмов или просмотр музыкальных видеоклипов на Web-узле	Пуск\Программы\Стандартные\Развлечения  Проигрыватель
Программа Звукозапись (рис. 2.36, д)	Служит для записи, наложения, воспроизведения и редактирования звукозаписей. Кроме того, звукозаписи можно связывать с другими документами или вставлять в них. Для использования программы Звукозапись на компьютере должны быть установлены звуковая плата и динамики. Для записи живого звука требуется также микрофон	Пуск\Все программы\Стандартные\Развлечения\Звукозапись  Звук - Звукозапись
Редактор WordPad	Предназначен для создания и редактирования текстовых файлов, содержащих форматирование или рисунки. Для редактирования простого текста или создания веб-страниц следует использовать блокнот	Пуск\Программы\Стандартные\ и выбрать  WordPad
Редактор личных символов	Используется для создания уникальных букв и эмблем для библиотеки шрифтов	Пуск\Выполнить, ввести eudcedit  Редактор личных символов
Сертификаты	Оснастка, помогающая запрашивать новые сертификаты общего ключа и управлять существующими сертификатами. Сертификаты используются различными службами безопасности общего ключа и приложениями, обеспечивающими проверку подлинности, целостность данных и безопасность коммуникаций по небезопасным сетям, таким как Internet. Администраторы могут управлять своими сертификатами, а также сертификатами других пользователей, компьютеров или служб. Пользователи могут управлять только своими сертификатами	
Системный монитор	Состоит из двух частей: собственно Системный монитор и Оповещения и журналы производительности	

Продолжение табл. 2.6

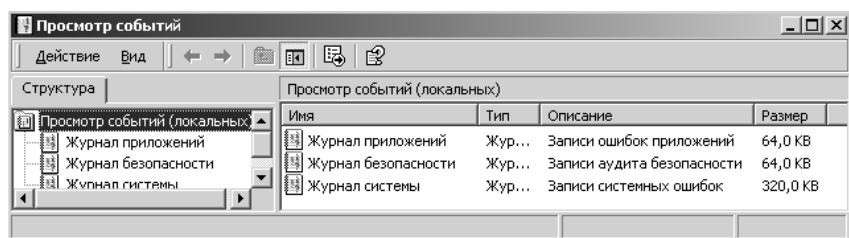
Программа	Функция	Запуск
Системный монитор	Системный монитор служит для сбора и просмотра в реальном времени данных памяти, диска, процессора, сети и других параметров в виде графика, гистограммы или отчета. С помощью компонента Оповещения и журналы производительности настраиваются журналы для записи данных представления и устанавливаются системные оповещения о том, что значение счетчика больше или меньше заданного ограничения	Пуск\Настройка\Панель управления\Администрирование\Системный монитор  монитор Системный монитор
Оснастка Службы	Помогает управлять службами компьютера, настраивать выполнение действий по восстановлению в случае сбоя службы, а также создавать пользовательские имена и описания для служб, чтобы в них было проще ориентироваться	Пуск\Настройка\Панель управления\Администрирование\Службы  Службы
Служба индексирования	Создает индексы содержимого и свойств документов на локальном жестком диске и на общих сетевых дисках. Имеется возможность контроля за включением сведений в индексы. Служба индексирования работает непрерывно и почти не нуждается в обслуживании	Управление компьютером\Служба индексирования  Служба индексирования
Windows Update	Сетевое интерактивное расширение Windows 2000, помогающее использовать компьютер наиболее эффективно. С помощью раздела Windows Update «Обновление приложений» можно проверить компьютер на наличие устаревших системных файлов и автоматически заменить их на последние версии	Запустить программу Windows Update. (http://windowsupdate.microsoft.com/) На основной странице щелкнуть Обновление приложений
Таблица символов	Используется для вставки в документы специальных символов, таких, как знак торговой марки, математические символы или символы из наборов символов других языков	Пуск\Программы\Стандартные\Служебные и Выбрать  Таблица символов
Программа Телефон	Используется для выполнения телефонных звонков или участия в видеоконференциях со своего компьютера. Для использования программы «Телефон» необходимо наличие подключенного к компьютеру телефона, модема, сетевой учетной записи, коммутатора, подключенного к локальной сети, или адреса Internet	Пуск\Программы\Стандартные\Связь и Выбрать  Телефон

Продолжение табл. 2.6

Программа	Функция	Запуск
Программа Управление компьютером	Помогает управлять локальным или удаленными компьютерами с помощью одной объединенной служебной программы рабочего стола. Она объединяет несколько средств администрирования Windows 2000 в одно дерево консоли, что обеспечивает легкий доступ к конкретным свойствам и средствам администрирования компьютера	Пуск\Настройка\Панель управления\Администрирование\Управление компьютером  Управление компьютером
Управление политикой безопасности IP	IPSec (Internet Protocol Security) в Windows является ключевой линией обороны на случай внутренних (частная сеть) и внешних (Internet, внешние сети) атак. IPSec шифрует данные, передаваемые между двумя компьютерами, защищая их от изменения и интерпретации при просмотре в сети. Управление IPSec осуществляется посредством конфигурации политики, создаваемой с помощью оснастки Управление политикой безопасности IP	Пуск\Выполнить, ввести <code>mmc</code> и нажать ОК\... В меню Консоль\Добавить/удалить оснастку, нажать кнопку Добавить\Управление политикой безопасности IP. Следовать инструкциям на экране
Консоль управления службой факсов	Помогает управлять факсимильными устройствами на локальном компьютере или на других компьютерах локальной сети. С помощью консоли управления службой факсов можно настроить разрешения безопасности, определить число звонков перед ответом факса, настроить устройство для приема факсов, а также задать приоритеты для отправки факсов	Пуск\Программы\Стандартные\Связь\Факс\Управление службой факсов  Управление...
Оснастка Шаблоны безопасности	Используется для создания и назначения шаблонов безопасности для одного или нескольких компьютеров. Шаблон безопасности — это файл настройки безопасности, которая может быть применена к локальному компьютеру или импортирована в объект Active Directory Групповая политика. При импорте шаблона безопасности в объект групповой политики оснастка Групповая политика обрабатывает этот шаблон и вносит соответствующие изменения для членов этого объекта групповой политики, которыми могут являться пользователи или компьютеры	
Экранная лупа	Является служебной программой отображения, облегчающей работу с экраном пользователям с плохим зрением. Данная программа выводит отдельное окно, в котором отображается увеличенная часть экрана.	Пуск\Программы\Стандартные\Специальные возможности\Экранная лупа  Экранная лупа

Окончание табл. 2.6

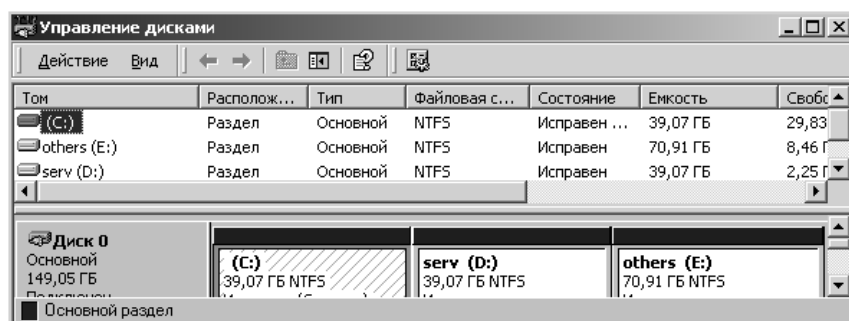
Программа	Функция	Запуск
Экранная лупа	Экранная лупа предназначена для облегчения работы людей с небольшими нарушениями зрения. Позволяет выполнять следующие действия — задание уровня увеличения; изменение размеров окна лупы; изменение положения окна лупы на рабочем столе; обращение экранных цветов; включение высококонтрастной схемы	Пуск\Программы\Стандартные\Специальные возможности\Экранная лупа  Экранная лупа
Программа HyperTerminal	Позволяет подключаться к другим компьютерам, узлам telnet, электронным доскам объявлений (BBS), интерактивным службам или ведомому компьютеру с помощью модема или нуль-модемного кабеля	Пуск\Программы\Стандартные\Связь\Выбрать  HyperTerminal
Программа Internet Explorer	При наличии подключения к Internet позволяет выполнять поиск и просмотр сведений в Internet. Для открытия нужной веб-страницы можно ввести ее адрес в адресной строке или выбрать из списка «Избранное». Internet Explorer также позволяет выполнять в Internet поиск людей, организаций и сведений по интересующим вопросам	Пуск\Программы\Internet Explorer  Internet Explorer
Программа NetMeeting	Предоставляет пользователям во всем мире новый способ общения, позволяет принимать участие в собраниях, работать в общих программах с помощью средств NetMeeting, а также передавать данные через Internet или интрасеть организации. Кроме того, можно разговаривать с другими людьми, используя аудио и видеоустройства и программу Разговор	Пуск\Программы\Стандартные\Связь\Выбрать  NetMeeting
Outlook Express	Программа электронной почты, позволяющая обмениваться с друзьями и коллегами электронными сообщениями, а также подписываться на группы новостей для обмена идеями и сведениями	Пуск\Программы\Outlook Express  Outlook Express
Программа Telnet	Обеспечивает поддержку протокола удаленного доступа Telnet, с помощью которого можно подключиться к удаленному компьютеру, сетевому устройству или частной сети TCP IP	Пуск\Выполнить\telnet Можно ввести telnet в командной строке:  C:\WINNT\system32\telnet.exe



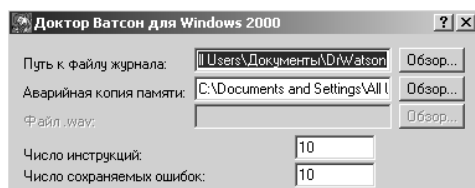
а



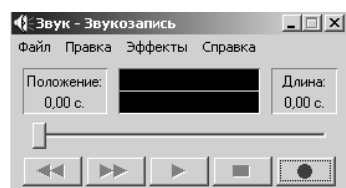
б



в



г



д

Рис. 2.36. Экранные формы некоторых служб и приложений Windows 2000:
 а — Просмотр событий; б — Проигрыватель DVD; в — Управление дисками;
 г — Доктор Ватсон; д — Звукозапись

Основные особенности Windows XP



Интерфейс Windows XP имеет модернизированный дизайн и содержит новые экраны с понятным интерфейсом, упрощенные меню и т. д.

Безопасность, простота в управлении. Windows XP Professional предоставляет безопасную файловую систему с возможностью шифрования важных данных. Связанная с безопасностью административная настройка гарантирует защиту и конфиденциальность системы.

Кроме того, в Windows XP Professional упрощено управление. Дружелюбный, модернизированный интерфейс Windows XP полностью обновлен. Он обладает более четкими линиями, насыщенными цветами, продуманной организацией, обеспечивающей быстрое перемещение и выполнение операций. В упрощенное меню кнопки Пуск автоматически добавляются часто используемые программы, которые могут быть запущены одним щелчком мыши. На обновленной панели задач файлы организуются «на лету». Мастер автоматической очистки поддерживает чистоту рабочего стола.

Улучшенная программа архивации с упрощенным управлением позволяет легко настроить регулярное копирование данных на магнитную ленту, диски Zip и другие носители данных.

Средство автоматического восстановления системы позволяет восстанавливать операционную систему в случае сбоя. Если по ошибке был удален важный файл или была загружена программа, которую не следовало загружать, то с помощью улучшенного средства Windows XP Восстановление системы можно «повернуть стрелки часов назад» и восстановить предыдущее состояние системы.

Windows XP Professional можно развертывать внутри организации постепенно. Обновление отдельных компьютеров по мере необходимости снижает стоимость введения новой технологии Windows XP и позволяет самостоятельно выбирать скорость обновления. Специализированные мастера предоставляют пошаговые инструкции для упрощения подключения новых устройств и запуска нового программного обеспечения.

Обновление офисных компьютеров, связанных программным обеспечением Windows 2000 Server, и управление ими могут осуществляться с помощью служб удаленной установки и групповых политик. Эти средства позволяют устанавливать, настраи-

вать и управлять несколькими компьютерами как группой, а не как отдельными машинами, а также наблюдать за их работой из одного места, что значительно снижает временные затраты и стоимость обслуживания.

Чтобы используемая версия Windows XP всегда была самой последней, следует пользоваться средством автоматического обновления для загрузки обновлений с веб-узла корпорации Майкрософт. Windows XP обеспечивает повышенную надежность и эффективность.

Защита данных компьютеров и сетей. В Windows XP появились новые мощные средства, разработанные для поддержки работоспособности сети при любых обстоятельствах. Сложное программное обеспечение защищает операционную систему каждого компьютера, а также создает защитный барьер, или бранд-мауэр, предотвращающий проникновение в сеть неавторизованных лиц и вирусов из Internet.

Устойчивость, надежность, совместимость. Windows XP Professional вводит новый стандарт программного обеспечения для бизнеса, объединяя производительность и надежность корпоративного уровня с беспрецедентной простотой эксплуатации. Построенная на фундаменте проверенной технологии Windows 2000 система Windows XP Professional включает все возможности Microsoft Windows XP Home Edition, а также новые и улучшенные средства, разработанные специально для бизнеса и опытных пользователей.

Время запуска снижено, а стабильность работы офисных программ выше, чем в любой предыдущей операционной системе.

Поддержка нескольких мониторов позволяет просматривать больше данных или одновременно выводить окна двух разных программ на компьютерах, оснащенных видеоадаптерами с двойным интерфейсом.

Технология Microsoft ClearType повышает читаемость содержимого экрана, особенно на переносных компьютерах.

Благодаря многоязыковой поддержке Windows XP Professional готова к использованию в любой стране мира. Возможность выбора любого из 24 языков исключительно удобна, если ваша компания имеет филиалы в разных странах и сотрудники, говорящие на разных языках, используют одни и те же компьютеры.

Для пользователей переносных компьютеров в Windows XP предусмотрены улучшенные средства управления электропитанием, продлевающие срок службы батарей. Как на перенос-

ных, так и стационарных компьютерах предоставляются более широкие возможности управления энергопотреблением компьютера.

Улучшенные средства связи Windows XP Professional предоставляют возможности управления последними технологиями в области мобильной вычислительной техники и беспроводной связи.

Дистанционное управление рабочим столом позволяет просматривать через Internet экран компьютера под управлением Windows XP Professional с любого другого компьютера под управлением Windows XP. Это означает, что можно «выйти на работу» с переносного компьютера, из дома или из офиса клиента в другой точке земного шара, как если бы вы были в двух разных местах одновременно.

Безопасные беспроводные подключения в Windows XP позволяют устанавливать связь и осуществлять совместную работу в реальном времени с использованием голосовой и видеосвязи, а также мгновенных сообщений. Нет необходимости тратить время на подключение — уведомление об обнаружении беспроводного устройства под управлением Windows XP в диапазоне связи выводится автоматически.

Windows XP Professional позволяет устанавливать одновременно до 10 подключений к Internet или других подключений с общим доступом к файлам. Маршрутизация сетей обеспечивает возможность одновременной установки беспроводных подключений, подключений Ethernet и по телефонной линии.

Быстрое переключение пользователей. Организовать работу нескольких пользователей на одном компьютере под управлением Windows XP проще, чем когда бы то ни было. Каждый пользователь компьютера может создать отдельную защищенную паролем учетную запись с личной настройкой и частными файлами. На одном компьютере могут быть активны сразу несколько учетных записей, переключение между ними осуществляется просто и быстро.

Например, если кто-то хочет просмотреть свою электронную почту, когда другой пользователь работает с компьютером, достаточно просто активизировать соответствующую учетную запись. Для этого не потребуются закрывать уже запущенные программы и завершать сеанс. Пользователи не смогут просмотреть файлы друг друга. Экран будет выглядеть в точности как перед переключением.

Использование мастеров. Чтобы настроить сеть дома или в небольшом офисе, необязательно быть экспертом в области организации сетей, *мастер* проведет через все необходимые для этого шаги. После ввода сети в эксплуатацию операционная система Windows XP помогает обслуживать ее, автоматически отслеживая изменения и корректируя параметры, чтобы обеспечить максимальную производительность при минимуме усилий со стороны пользователя.

Улучшенная справка по каждой задаче. В Windows XP имеются все необходимые сведения о самой операционной системе и компьютере. Новый центр справки и поддержки в Windows XP позволяет быстро получить следующие сведения.

- понятные инструкции по выполнению конкретных действий;
- исчерпывающие статьи на интересующие темы;
- советы по устранению неполадок;
- бесплатные обновления продуктов.

Встроенные средства открытия совместного доступа и организации фотографий. В Windows XP максимально упрощены просмотр фотографий и открытие совместного доступа к ним для друзей, членов семьи и коллег. Достаточно подключить цифровую камеру к компьютеру и немедленно будут отображены фотографии, готовые к редактированию, отправке по электронной почте или печати. Предоставляется также возможность опубликовать их непосредственно на веб-узле или сохранить на компакт-диске

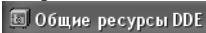
Воспроизведение музыки, видеозаписей и компакт-дисков. В состав Windows XP входит проигрыватель Windows Media, который позволяет:

- слушать Internet-радио;
- загружать музыку и фильмы из Internet;
- смотреть потоковые видеозаписи;
- воспроизводить собственные компакт-диски.

Проигрыватель Windows Media позволяет создать собственный настраиваемый *дисковод со сменой дисков*. Компьютер будет воспроизводить записи в любом заданном порядке, в любое время.

В табл. 2.7 (см. также рис. 2.37) приводятся основные функции, отличающие XP от более ранних версий ОС (отметим, что некоторые из них, хотя они и не вошли в табл. 2.6, были реализованы в более поздних выпусках Windows 2000).

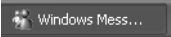
Таблица 2.7. Программы, службы и функции, встроенные в Windows XP

Программа	Функция	Запуск
Диспетчер служебных программ	Позволяет проверять состояние программ специальных возможностей, а также запускать их и останавливать. Встроенными программами, доступными из диспетчера служебных программ, являются Экранная лупа, Экранный диктор и Экранная клавиатура. С помощью диспетчера служебных программ можно настроить Windows на автоматический запуск программ специальных возможностей при каждом входе в систему, закреплении элементов рабочего стола или запуске диспетчера служебных программ. Например, можно указать, что программа Экранная лупа должна запускаться автоматически при входе в систему, что позволит каждый раз при входе в систему пропускать все шаги для открытия программы Экранная лупа	Чтобы открыть диспетчер служебных программ, следует нажать клавиши <  + U > 
Планировщик заданий	Позволяет назначить выполнение сценария, запуск программы или открытие документа на наиболее удобное время — назначать задание к выполнению ежедневно, еженедельно, ежемесячно или в определенные моменты (например, при запуске системы); изменять расписание для задания; прекращать выполнение назначенного задания; настраивать режим запуска задания в назначенное время	Планировщик заданий запускается при каждом запуске Windows XP и выполняется в фоновом режиме
Программа связи с данными	Используется для создания и управления подключениями между локальным компьютером и хранилищами данных OLE DB. При создании связи с данными заполняется файл .udl, который можно использовать для будущих подключений к базе данных	Windows проводник\Выбрать папку, в которой требуется сохранить UDL-файл\Выбрать пункт Создать, а затем Текстовый документ\Ввести нужное имя файла (например, mydata.udl), а затем нажать <Enter>
Программа Общие ресурсы DDE	Позволяет управлять обменом данными между программами и совместным использованием данных этими программами в сети путем создания, изменения и удаления общих ресурсов DDE (Dynamic Data Exchange)	Пуск\Выполнить\В поле Открыть ввести ddeshare 

Продолжение табл. 2.7

Программа	Функция	Запуск
Поиск и устранение ошибок на диске	Для обнаружения ошибок файловой системы и поврежденных секторов на жестком диске можно использовать служебную программу проверки диска. Перед запуском проверки диска следует закрыть все файлы на нем. Если том используется, на экран будет выведено сообщение с предложением выполнить проверку диска после перезагрузки системы. Во время проверки диск недоступен для выполнения других задач. При использовании файловой системы NTFS Windows выполняет запись всех транзакций для файлов, автоматически заменяет поврежденные кластеры и сохраняет на томе NTFS копию наиболее важных данных о каждом из файлов	Мой компьютер\выбрать локальный диск, который требуется проверить\файл\Свойства\На вкладке Сервис в группе Проверка диска нажать кнопку Выполнить проверку
Окно просмотра изображений	Позволяет просматривать и поворачивать изображения, а также выполнять основные задачи с документами изображений, включая факсы, без открытия графического редактора. Это окно позволяет переносить на компьютер изображения с цифровой камеры или сканера и просматривать их в виде эскизов. Кроме того, можно просматривать изображения в режиме слайд-шоу и добавлять пометки в факсы. Если для хранения рисунков используется папка Мои рисунки, изображения отображаются автоматически в виде эскизов или как диафильм. Режимы представления изображений в папке Мои рисунки можно выбрать с помощью команд меню Вид	Пуск\Мои рисунки\Дважды щелкнуть изображение, чтобы открыть его в окне просмотра изображений  Мои рисунки
Компонент Источники данных (ODBC) (рис. 2.37, а)	Используется для доступа к данным из различных систем управления базами данных. Например, при наличии программы, работающей с данными из базы данных SQL, компонент Источники данных (ODBC) позволит использовать эту же программу для доступа к данным в базе данных Visual FoxPro. Для этого необходимо добавить в систему программные компоненты (драйверы). Компонент Источники данных (ODBC) помогает добавить и настроить эти драйверы	Пуск\Панель управления\Дважды щелкнуть значок Администрирование\Дважды щелкнуть значок Источники данных (ODBC)  Администратор источников

Продолжение табл. 2.7

Программа	Функция	Запуск
Служба Windows MSN Messenger Service (рис. 2.37, б)	Служба позволяет — видеть, кто находится в интерактивном режиме; оперативно отправлять сообщения; вызывать компьютеры нужных пользователей; отправлять файлы; обмениваться сообщениями с группой друзей	Если программа Windows Messenger еще не открыта, это можно сделать, нажав кнопку Пуск и выбрав пункты Все программы и Windows Messenger 
Мастер переноса файлов и параметров	Позволяет перемещать файлы данных и личные настройки со старого компьютера на новый без необходимости повторения на новом компьютере действий по настройке, выполненных на старом компьютере. Например, со старого компьютера на новый можно перенести личные свойства экрана, параметры папок и панели задач, настройки обозревателя Internet и электронной почты. Данный мастер также перемещает некоторые файлы или целые папки, такие как Мои документы, Мои рисунки и Избранное. При перемещении свойств программ с помощью мастера переноса файлов и параметров переноса паролей не происходит. Эта функциональная возможность мастера переноса файлов и параметров помогает сохранить конфиденциальность паролей	Пуск\ Все программы\ Стандартные\ Служебные\ Мастер переноса файлов и параметров 
Средства администрирования на консоли управления Microsoft	Консоль управления MMC (Microsoft Management Console) содержит средства администрирования (оснастки), которые используются для администрирования компьютеров, служб, других системных компонентов и сетей	Открыть консоль управления MMC. В меню Консоль выбрать Добавить оснастку. В диалоговом окне Добавить\удалить оснастку, нажать кнопку Добавить\Выбрать оснастку, которую требуется добавить на консоль, и нажать кнопку Добавить
Программа Windows Movie Maker (рис. 2.37, в)	Позволяет перемещать записанный звук и видео с источника (например, с аналоговой или цифровой видеокамеры) на компьютер. Помимо использования записанного содержимого, для создания фильмов можно импортировать существующие звуковые и видеофайлы.	Пуск\Все программы\Стандартные\ Windows Movie Maker 

Продолжение табл. 2.7

Программа	Функция	Запуск
	Созданный фильм можно продемонстрировать семье и друзьям непосредственно из Windows Movie Maker, отправив его по электронной почте или разместив на Web-узле	
Средство Восстановление системы	Компонент системы Windows XP Professional, с помощью которого при возникновении проблем можно восстановить предыдущее состояние компьютера без потери личных файлов (таких, как документы Microsoft Word, перечень просмотренных страниц, рисунки, избранные файлы и сообщения электронной почты). Программа Восстановление системы ведет наблюдение за изменениями системы и некоторыми файлами приложений и автоматически создает легко идентифицируемые точки восстановления, которые позволяют вернуть систему к состоянию на данный момент времени. Восстановление никак не влияет и не изменяет личные файлы данных	Точки восстановления создаются ежедневно, а также во время существенных системных событий (таких, как установка приложения или драйвера). Пользователь также имеет возможность в любое время создавать именованные точки восстановления
Диспетчер задач Windows	Предоставляет сведения о производительности компьютера, а также о выполняемых компьютером программах и процессах. При наличии подключения к сети можно также просматривать состояние сети и результаты ее функционирования. В зависимости от сетевой среды, при использовании компьютера совместно с другими пользователями можно просматривать дополнительные сведения об этих пользователях. С помощью диспетчера задач Windows можно завершать выполнение программ или процессов, запускать программы, а также просматривать динамику производительности компьютера	Чтобы открыть диспетчер задач Windows, следует щелкнуть правой кнопкой пустое место на панели задач и выбрать в контекстном меню команду Диспетчер задач 
Регулятор громкости	Служит для регулировки громкости, баланса и тембра звуков, воспроизводимых с помощью компьютера или приложений мультимедиа. Кроме того, регулятор громкости можно использовать для регулирования уровня системных звуков, микрофонов, компакт-дисков, линейного входа, синтезатора и волнового	Пуск\Все программы\Стандартные\Развлечения\Регулятор громкости 

Продолжение табл. 2.7

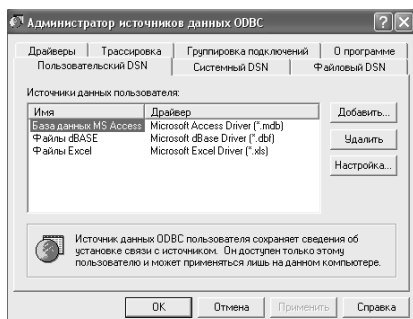
Программа	Функция	Запуск
Регулятор громкости	выхода. При наличии звуковой платы регулятор громкости можно использовать для управления громкостью и балансом каналов при воспроизведении звуковых файлов	Пуск\Все программы\Стандартные\Развлечения\Регулятор громкости 
Экранная клавиатура (рис. 2.37, г)	Приложение, отображающее виртуальную клавиатуру на экране и позволяющее людям с ограниченной подвижностью печатать на экране с помощью указателя мыши или джойстика. Экранная клавиатура предназначена для облегчения работы людей с ограниченной подвижностью. Окно программы, в которое вводятся символы с помощью экранной клавиатуры, должно быть активным. Экранная клавиатура поддерживает три режима ввода данных — режим использования кнопок мыши, при котором нужно щелкнуть выбранные клавиши кнопкой мыши; режим сканирования, при котором на экранной клавиатуре выделяются области, в которых можно вводить символы путем нажатия сочетания клавиш или используя кнопочное устройство ввода, режим ожидания, при котором в течении времени ожидания символ выделяется указателем мыши или джойстиком и по истечении этого времени печатается автоматически	1. Пуск\Все программы\Стандартные\Специальные возможности\Экранная клавиатура  2. Нажать клавиши <Win> + U> при отображении диалогового окна Операционная система Windows во время запуска Windows
Экранный диктор	Является программой озвучивания текста для пользователей с частичной и полной потерей зрения. Программа озвучивает то, что отображено на экране — содержимое активного окна, параметры меню или введенный текст. Программа Экранный диктор была разработана для совместного использования с программами Блокнот, WordPad, Internet Explorer, рабочим столом Windows и программой установки Windows. В других программах Экранный диктор может озвучивать слова неправильно. Программа Экранный диктор доступна не для всех языков и поддерживается только англоязычной версией Windows XP	Пуск\Все программы\Стандартные\Специальные возможности\Экранный диктор 

Продолжение табл. 2.7

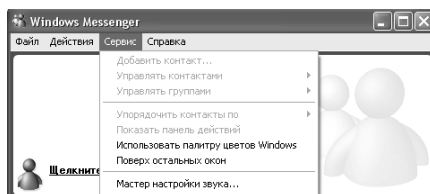
Программа	Функция	Запуск
Оснастка Службы компонентов	Используется системными администраторами для внедрения и администрирования приложений служб компонентов через графический интерфейс пользователя, а также для автоматизации административных задач с помощью языков программирования и подготовки сценариев. Разработчики программного обеспечения могут использовать службы компонентов для наглядной настройки стандартных действий компонентов и программ, например безопасности и участия в операциях, а также для интеграции компонентов в приложения служб компонентов	Пуск\Панель управления\Дважды щелкнуть значок Администрирование\Дважды щелкнуть значок Службы компонентов 
Работа с очередью сообщений	Очередь сообщений (также используется название MSMQ) представляет собой инфраструктуру и среду для выполнения распределенных, свободно подключаемых приложений обмена сообщениями. Приложения очереди сообщений на исходных компьютерах отправляют сообщения в очереди, в которых сообщения временно хранятся, а приложения на конечных компьютерах получают сообщения из очередей. Такие приложения могут производить обмен данными в неоднородных сетях и с компьютерами, которые могут оказаться временно отключенными от сети. Для управления очередью сообщений используется несколько оснасток консоли MMC	Открыть окно Установка и удаление программ\Нажать кнопку Добавление и удаление компонентов Windows\В окне установить флажок Очередь сообщений. Если переключатели очереди сообщений устанавливать не требуется, снять флажок Переключатели
Диспетчер устройств	Предоставляет сведения об установленном на компьютере оборудовании и его настройках, а также о взаимодействии этого оборудования с программами компьютера. С помощью диспетчера устройств можно обновлять драйверы устройств для установленного на компьютере оборудования, изменять настройки оборудования, а также устранять неполадки	Пуск\Панель управления\Дважды щелкнуть значок Система\На вкладке Оборудование нажать кнопку Диспетчер устройств 

Окончание табл. 2.7

Программа	Функция	Запуск
Компонент Сведения о системе	Собирает и отображает сведения о конфигурации системы. Сотрудникам служб технической поддержки для устранения неполадок в системе требуются определенные сведения о компьютере. Компонент Сведения о системе позволяет быстро найти нужные сведения	Пуск\Выполнить\В поле Открыть ввести msinfo32.exe 
Консоль MMC (Microsoft Management Console)	Средство создания, сохранения и открытия наборов средств администрирования, называемых консолями. Консоли содержат такие элементы, как оснастки, расширения, элементы управления, задачи, мастера и документацию, необходимую для управления многими аппаратными, программными и сетевыми компонентами системы Windows. Можно добавлять элементы в существующую консоль MMC или можно создавать новые консоли и настраивать их для управления конкретными компонентами системы	Пуск\Выполнить\В поле Открыть ввести mmc 
Оснастка Управляющий элемент WMI	Позволяет настраивать параметры WMI на удаленном компьютере или локальном компьютере. С помощью оснастки Управляющий элемент WMI можно задавать разрешения для прошедших проверку пользователей и групп, включать или выключать запись журнала ошибок, создавать резервные копии хранилища WMI или выполнять другие задачи по настройке	Пуск\Выполнить\В поле Открыть ввести wmiimgmt.msc 
Запуск MS DOS	MS DOS является операционной системой с интерфейсом командной строки. Режим MS DOS — это оболочка, в которой в 32-битовых системах, таких, как Windows, эмулируется среда MS DOS. Программы MS DOS могут работать под Windows, а также создавать файл описания программы (PIF), который появляется в виде ярлыка на рабочем столе	Пуск\Все программы\Стандартные\Командная строка. Для завершения сеанса MS DOS ввести exit в окне командной строки 



а



б



в



г

Рис. 2.37. Экранные формы некоторых служб и приложений Windows XP:
 а — Администратор источников данных; б — Windows Messenger; в — Windows movie maker; г — Экранная клавиатура

Windows Vista



Является следующей версией ОС Windows, которая официально была выпущена 30 января 2007 г. Утверждают, что она представляет собой 6-ю версия Windows NT (Windows NT 5 вышла под названием Windows 2000, а Windows XP — версия 5.1).

ОС распространяется в следующих вариантах:

- Windows Vista Starter — стартовая версия;
- Home Basic/Home Premium — версии для домашнего использования;
- Business — бизнес-версия;
- Vista Enterprise — корпоративная;
- Ultimate — максимальный набор функций.

Windows Vista вышла на рынок больше чем через пять лет после появления Windows XP, поэтому XP может считаться самой «долгоживущей» ОС Microsoft. Основной чертой Windows Vista является повышение безопасности, поскольку предыдущая версия была довольно уязвима перед атаками вирусов и другого вредоносного ПО. Windows Vista содержит несколько уровней защиты, включая улучшенный встроенный центр безопасности и брандмауэр, а также утилиту для запрета выполнения нежелательных программ. Приведенные ниже иллюстрации базируются на Windows Vista Business.

Интерфейс Windows Vista в целом организован традиционным образом. Рабочий стол (рис. 2.38) — главная область экрана, которая появляется после включения компьютера и входа в ОС, он включает панель задач и боковую панель.

На панели задач отображаются значки запущенных программ, позволяющих переключаться между ними. Она также содержит кнопку **Пуск** , которой можно воспользоваться для доступа к программам, папкам и параметрам компьютера. Сбоку экрана расположена боковая панель, которая содержит так называемые мини-приложения.

Кнопка и меню Пуск. Чтобы открыть меню **Пуск**, следует нажать кнопку  **Пуск** в левом нижнем углу экрана (или же клавишу с эмблемой Windows  на клавиатуре).

Меню (рис. 2.39) состоит из трех основных разделов:

- основное меню — на большой панели слева отображается краткий список программ и файлов компьютера (на рис. 2.39, 1), этот список «отфильтрован» через поле поиска;



Рис. 2.38. Общий вид рабочего стола Windows Vista:

1 — ярлыки приложений; 2 — панель задач; 3 — боковая панель (мини-приложения); 4 — предварительный просмотр окон приложений; 5 — трехмерное перелистывание окон приложений

- поле поиска (рис. 2.39, 2), позволяющее выполнять поиск программ и файлов на компьютере с помощью введения ключевых слов (в данном случае введенная подстрока win вызывает отображение в основном меню только соответствующих приложений и файлов);
- правая панель (рис. 2.39, 4) предоставляет доступ к наиболее часто используемым папкам, файлам, параметрам и ключевым свойствам системы. Здесь же можно завершить сеанс работы пользователя в Windows или выключить компьютер (рис. 2.39, 6).

Поле Поиск. Для его использования следует открыть меню Пуск и начать ввод текста. При вводе текста результаты поиска отображаются над полем поиска в левой области меню Пуск, причем программа, файл или папка будет отображена в результатах поиска в следующих случаях:

- какое-либо слово в названии совпадает с введенной подстрокой критерия поиска или начинается с нее;

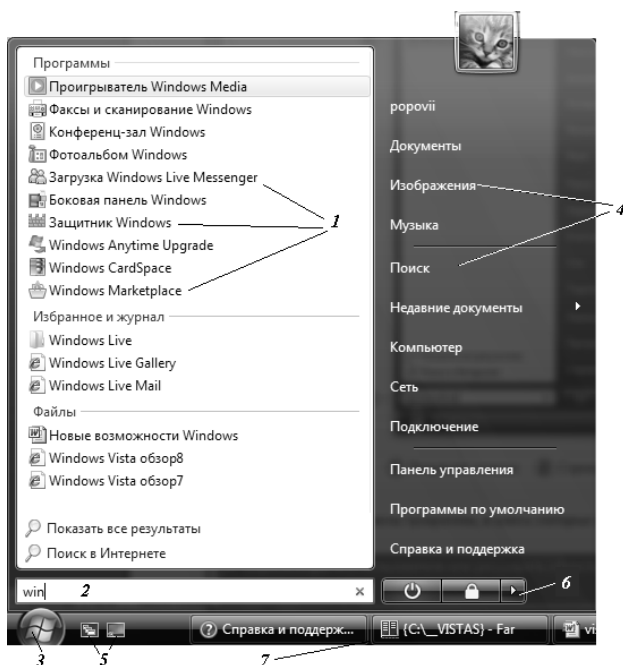


Рис. 2.39. Основные элементы меню Пуск:

1 — основное меню; 2 — поле поиска; 3 — кнопка Пуск; 4 — правая панель; 5 — область быстрого запуска (кнопки управления окнами); 6 — кнопки питания и блокировки; 7 — область активных приложений

- любой текст содержимого файла (например, текст в текстовом редакторе) совпадает с введенной подстрокой критерия поиска или начинается с нее;
- любое слово в свойстве файла, например автор, совпадает с введенным критерием поиска или начинается с него.

Кроме программ, файлов, папок и контактов поиск также выполняется в Избранном обозревателя Internet и в списке посещенных Web-узлов. Если какие-нибудь из Web-страниц содержат поисковое слово, они будут отображены под заголовком Избранное и журнал.

Правая панель меню Пуск содержит ссылки на часто используемые компоненты:

- личная папка, названная именем текущего пользователя Windows, которая содержит личные файлы пользователя, включая папки Документы, Музыка, Изображения и Видео;

- Документы, Изображения, Музыка — папки, в которых можно сохранять и открывать текстовые файлы, электронные таблицы и другие типы документов; сохранять и просматривать рисунки и графические файлы; сохранять и воспроизводить музыку и другие аудиофайлы, соответственно;
- Игры — папка, в которой осуществляется доступ ко всем играм на компьютере;
- Поиск — открывает окно для выполнения поиска данных на компьютере с расширенными параметрами;
- Недавние документы — открывает список недавно использовавшихся файлов;
- Компьютер — окно, предоставляющее доступ к дискам, камерам, принтерам, сканерам и другому подключенному к компьютеру оборудованию;
- Сеть — окно, предоставляющее доступ к компьютерам и устройствам в сети;
- Подключение — окно для создания подключения к новой сети;
- Панель управления — диалог, предоставляющий возможность изменить внешний вид и функциональность компьютера, добавлять и удалять программы, устанавливать сетевые подключения и выполнять управление учетными записями пользователей;
- Программы, используемые по умолчанию, — окно для настройки программ Windows, используемых по умолчанию для работы в Internet, редактирования изображений, отправки электронной почты и воспроизведения музыки и видео;
- Справка и поддержка — открывает центр справки и поддержки Windows, в котором можно найти справочные сведения по использованию Windows и компьютера.

Внизу правой панели находятся две кнопки (рис. 2.39, б): кнопка питания (для выключения компьютера) и кнопка блокировки (для блокировки компьютера без его выключения). После блокировки компьютер не может быть использован без предварительной разблокировки с помощью пароля. При щелчке стрелки рядом с кнопкой блокировки будет отображено меню с дополнительными параметрами, такими как переключение пользователей, завершение сеанса пользователя, перезагрузка и завершение работы компьютера.

На рис. 2.40 приведен пример экрана содержимого каталога Компьютер\Локальный диск C:\Picts. Здесь показана одна из

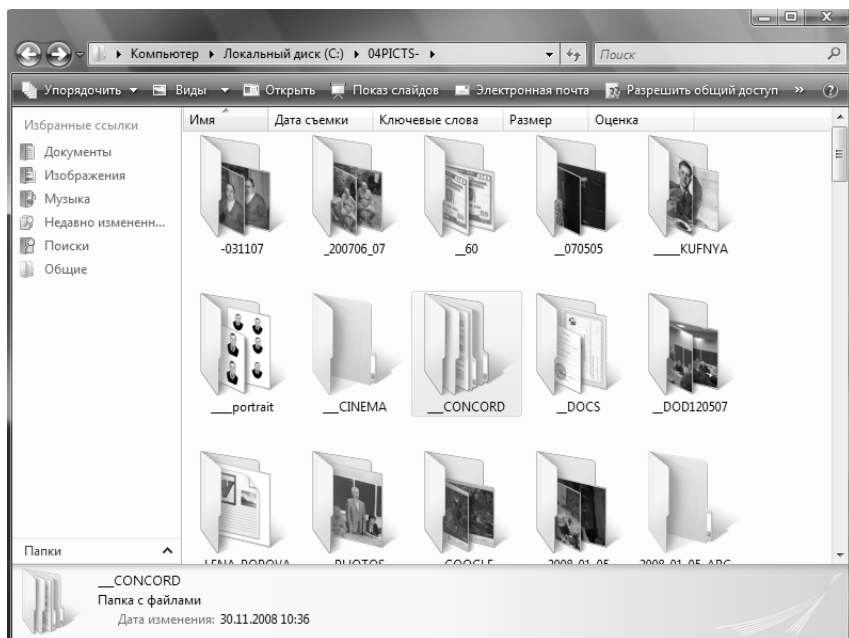


Рис. 2.40. Окно просмотра содержания каталога

новых возможностей интерфейса — манипулирование трехмерными образами объектов. Файлы и папки в Vista Explorer являются трехмерными объектами, позволяющими видеть эскизы их содержания.

Боковая панель Windows — длинная вертикальная панель на правом краю рабочего стола (см. рис. 2.38, 3). Она содержит мини-приложения (мини-программы), предоставляющие короткую справку или быстрый доступ к часто используемым средствам. Мини-приложения позволяют, например, показывать слайды, просматривать постоянно обновляемые заголовки новостей и искать контакты.

Windows поставляется с небольшим набором мини-приложений, но лишь некоторые из них появляются на боковой панели по умолчанию (рис. 2.41, а). На боковой панели необходимо нажать значок плюс (+), чтобы открыть коллекцию мини-приложений (рис. 2.41, б).

В качестве примера рассмотрим мини-приложение Часы. При указании на ярлык мини-приложения в его верхнем правом углу появляются две кнопки: **Заккрыть** (рис. 2.41, в, 1) и **Пара-**

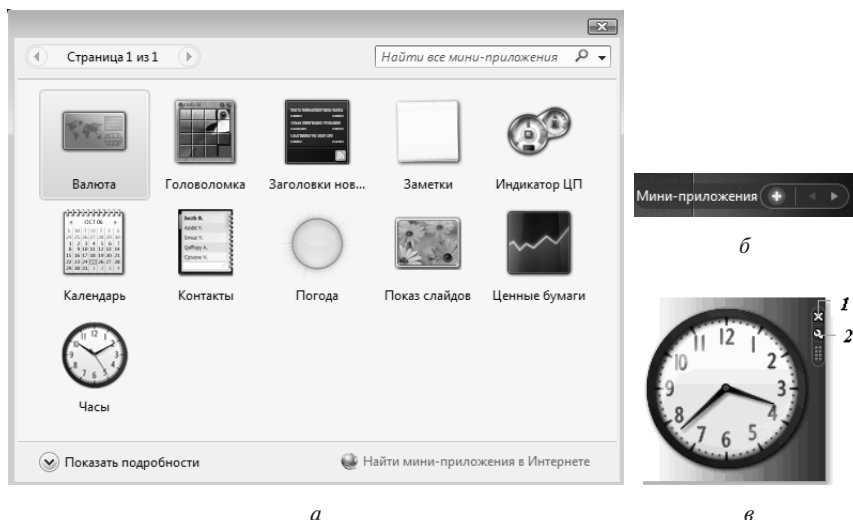


Рис. 2.41. Мини-приложения Windows Vista:

а — коллекция мини-приложений; *б* — кнопка управления мини-приложениями;
в — Часы (кнопки Закреть (1) и Параметры (2))

метры (рис. 2.41, *в*, 2). Щелчок кнопки Закреть удаляет мини-приложение, а кнопки Параметры — вызывает возможности модификации обозначения часов, изменения часового пояса и отображения секундной стрелки.

Панель задач представляет собой горизонтальную полосу в нижней части экрана (см. рис. 2.38, 2), которая состоит из четырех основных частей:

- кнопки Пуск , открывающей меню Пуск (см. рис. 2.39);
- панели быстрого запуска, позволяющей запустить программу нажатием кнопки мыши (см. рис. 2.39, 5);
- области, которая отображает открытые программы и документы и дает возможность быстро переключаться между ними (см. рис. 2.39, 7);
- области уведомлений, в которой находятся часы и значки, отображающие состояние некоторых программ и параметров компьютера.

При наведении указателя мыши на кнопку панели задач появляется маленькая картинка с эскизом (миниатюрным изображением) соответствующего окна (см. рис. 2.38, 4), что особенно полезно в тех случаях, когда нельзя идентифицировать окно по одному

лишь заголовку. Если в окне воспроизводится видео или анимация, можно будет увидеть это и на предварительном изображении.

Windows Aero. Появившиеся в Windows Vista трехмерные визуальные эффекты (см. рис. 2.40 и рис. 2.38, 5) доступны, если на компьютере установлена тема оформления Windows Aero (трактруется как тема рабочего стола).

Стиль Windows Aero (в некоторых источниках упоминается как AeroGlass, буквально — «Воздушная линза») обеспечивает в Windows Vista визуальные эффекты высокого качества. В его основе лежит дизайн с эффектом прозрачного стекла, плавной анимацией окон и новыми цветами. Частью оформления Windows Aero является трехмерное перелистывание (см. рис. 2.38, 5), позволяющее отображать обновляющиеся эскизы открытых окон и кнопок панели задач в трехмерном пространстве.

Для выполнения Windows Aero необходима следующая минимальная конфигурация ПК:

- 32-разрядный (x86) или 64-разрядный (x64) процессор с частотой 1 ГГц;
- 1 Гбайт оперативной памяти (RAM);
- видеокарта 128 Мбайт.

Для работы Aero также необходим графический процессор с поддержкой DirectX 9, Windows Display Driver Model Driver, Pixel Shader 2.0 на уровне оборудования и 32 разряда на пиксель.

Центр начальной настройки (Welcome Center) в принципе позволяет пройти через все пункты один раз сразу после установки Windows. Открыв данный диалог (рис. 2.42), можно получить доступ к ряду служебных функций. Вот некоторые из них:

- Просмотр сведений о компьютере (View Computer Details). Отсюда можно запустить диспетчер устройств, изменить настройки удаленного доступа, отрегулировать защиту системы и дополнительные настройки. Все это доступно через окно свойств системы, которое можно вызывать, если нажать правой клавишей мыши на Компьютер (вместо Мой Компьютер в Win XP) и выбрать Свойства;
- Подключение к Internet (Connect to the Internet). Мастер подключения позволит настроить коммутируемое соединение, выделенную линию PPPoE (Point-to-Point over Ethernet для модемов DSL или кабельного) или выход через беспроводную сеть. Если у Vista есть необходимые драйверы, то настраивать беспроводное соединение будет намного легче, чем в Windows XP;

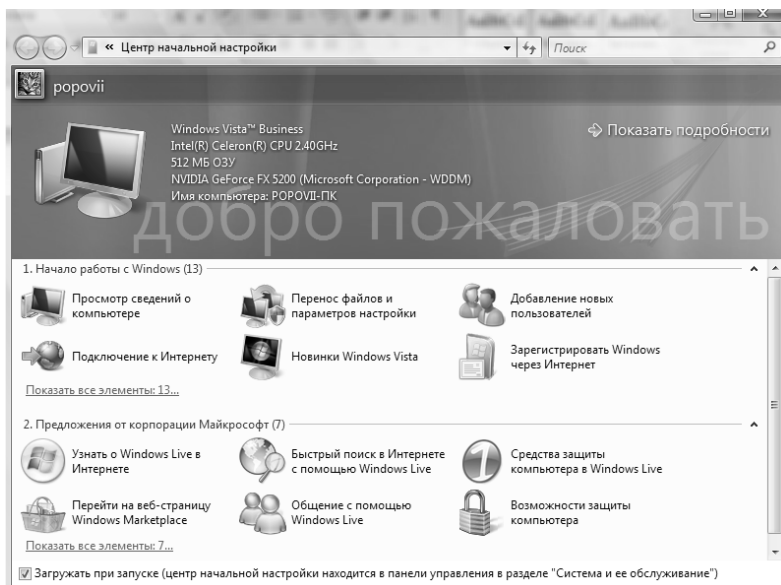


Рис. 2.42. Экран Центра начальной настройки

- Новинки Windows Vista (What's New in Windows Vista). Пункт запускает помощь Windows, где рассказывается о новых функциях (табл. 2.8), а также другие информационные разделы.

Таблица 2.8. Новые возможности Windows Vista Business

Раздел	Содержание
Поиск и упорядочение 	В правом верхнем углу каждой папки имеется поле поиска. При вводе данных в поле поиска осуществляется отбор отображаемых элементов на основе введенной информации (слова в имени файла, метаданные файла и другие его свойства). Если расположение файла неизвестно или необходимо выполнить расширенный поиск не только по имени файла или его отдельному свойству, можно использовать также Папки поиска
Центр архивации и восстановления 	Центр архивации и восстановления упрощает резервное копирование параметров, файлов и программ в любое место и время по выбору пользователя, с удобным автоматическим планированием. Можно создать резервную копию на компакт-диске или DVD-диске, внешнем жестком диске, другом жестком диске своего компьютера, флэш-памяти либо на другом компьютере или сервере, подключенном к сети

Продолжение табл. 2.8

Раздел	Содержание
Новая улучшенная справка	Справка и поддержка полностью обновлена для данной версии. Она позволяет получать быстрые ответы на общие вопросы, предлагает способы выявления неисправностей и инструкции по выполнению тех или иных действий. При наличии подключения к Internet можно гарантировать получение самых новых и последних версий всех разделов справки
Безопасность 	Такие возможности, как брандмауэр Windows и Защитник Windows помогают повысить безопасность компьютера. Центр обеспечения безопасности Windows имеет ссылки, которые позволяют выполнить проверку брандмауэра компьютера, антивирусного программного обеспечения и состояние обновления. Контроль учетных записей (UAC) препятствует выполнению несанкционированных изменений на компьютере, запрашивая разрешение перед выполнением действий, которые могут повлиять на работу компьютера, и перед изменением параметров, оказывающих влияние на других пользователей
Сеть 	С использованием центра сети и общего доступа можно получить информацию о состоянии сети в реальном времени и ссылки на особые операции. Появляется возможность создания более безопасной беспроводной сети, надежного подключения к общедоступным сетям в пунктах подключения к беспроводной сети и контроля безопасности сети. Реализован простой доступ к файлам и общим сетевым устройствам, таким как принтеры, и использование интерактивной диагностики для выявления и решения сетевых проблем
Синхронизация, общий доступ 	Синхронизация с другими устройствами, такими как портативные музыкальные проигрыватели и мобильные устройства Windows. Центр синхронизации позволяет синхронизировать устройства, управлять синхронизацией устройств, запускать ручную синхронизацию, узнавать состояние текущих процессов синхронизации и проверять возможность конфликтов. При общем доступе к файлам и папкам другие пользователи могут открывать и просматривать их, как если бы эти файлы и папки находились на их собственном компьютере. Пользователи также могут вносить изменения при наличии разрешений
Internet Explorer 	Web-каналы, просмотр web-страниц с использованием вкладок и всегда доступный поиск — это лишь некоторые новые возможности Internet Explorer. Web-каналы предоставляют часто обновляемое содержимое, публикуемое web-узлом, на который можно подписаться для автоматической доставки web-обозревателю пользователя. С помощью канала можно получать такое содержимое, как экстренные выпуски новостей или обновления для блога, без посещения web-узла. Просмотр с использованием вкладок позволяет открывать множество web-страниц в одном окне обозревателя. Web-страницы или ссылки можно открывать на новых вкладках и переключаться между ними, щелкая вкладки

Окончание табл. 2.8

Раздел	Содержание
Windows Meeting Space 	Совместная работа и распределение документов между пользователями в Internet, совместное использование рабочего стола или любой программы пользователя с другими участниками сеанса связи, распределение и совместное редактирование документов и обмен заметками
Возможности мобильных ПК 	С помощью центра мобильной связи можно настроить параметры, которые регулярно меняются при перемещении пользователя с одного места к другому (например, громкость и яркость экрана), и проверить состояние подключения. Используйте вторичный или вспомогательный экран, чтобы узнавать об очередной запланированной встрече, читать сообщения электронной почты, слушать музыку или просматривать новости, не открывая мобильный ПК. Вспомогательный экран также может быть у таких устройств, как сотовый телефон или телевизор
Возможности планшетных ПК 	Улучшение распознавания рукописного ввода с помощью персональной настройки распознавателя. Использование жестов пера для перемещения и выполнения типовых действий. Благодаря оптимизированному указателю более четко видны действия пера. Использование панели ввода для рукописного ввода или экранной клавиатуры в любом месте экрана. Использование сенсорного экрана для выполнения действий с помощью пальца (сенсорный экран доступен только в сенсорных планшетных ПК)
Изображения 	Папка Изображения и фотоальбом Windows упрощают просмотр, систематизацию, редактирование, совместное использование и печать цифровых изображений. Подключив цифровую камеру к компьютеру, можно автоматически перенести фотографии в папку Изображения. Так с помощью фотоальбома Windows можно обрезать фотографии, устранять эффект «красных глаз» и выполнять коррекцию цветов и поддержки
Центр специальных возможностей 	Новый центр специальных возможностей заменяет компонент Специальные возможности в более ранних версиях Windows. Центр специальных возможностей имеет ряд преимуществ и новых функций, в том числе централизованный доступ к специальным возможностям и опросный лист, который можно использовать для получения полезных советов по специальным возможностям
Ресурсы для малого бизнеса 	Новые средства в Windows Vista Business помогут упростить выполнение задач, сделать работу эффективнее и защитить важную деловую информацию. Обеспечен доступ к дополнительным сведениям о средствах работы в сети, архивировании, использовании факсов и сканеров, включенных в Windows Vista Business, о быстром доступе к бесплатным деловым и технологическим ресурсам в Internet для малого бизнеса

Windows 7 (Vienna)



Windows 7 является развитием Windows Vista (которая имела номер версии Windows 6.0). Кодовое название версии — Vienna. Ядро Windows 7 основывается на ядре Windows Server 2008 (которое представляет собой модификацию ядра Windows Vista). Из этого следует, что новая операционная система совместима со всем аппаратным обеспечением, приложениями и драйверами, которые были написаны для Windows Vista. Таким образом, переход на новую систему не будет таким болезненным, как это было с выходом Windows Vista.

Согласно официальной информации, для запуска Windows 7 требуется компьютер, оборудованный процессором с частотой не ниже 1 ГГц, как минимум 1 Гбайт оперативной памяти и графическим контроллером с поддержкой программного интерфейса DirectX 9.

Интерфейс Windows 7. Значительное количество изменений и доработок пришлось на интерфейс Windows 7. На рис. 2.43 приведен пример окна интерфейса, иллюстрирующего основные особенности Windows 7 (на примере пробной версии «Release Candidate, build 7100» — RC 7100).

Прежде всего, комплекс особенностей Windows Aero (прозрачность и всплывание окон, трехмерное перелистывание и пр.) теперь не требует специальной установки в качестве темы рабочего стола, а непосредственно встроен по умолчанию. Любое приложение может быть помещено в панель задач и быть всегда под рукой.

Высота панели задач увеличена на десять пикселей, панель сделана полупрозрачной и объединена с панелью быстрого запуска (рис. 2.43, 2), значки приложений укрупнены, и теперь они соответствуют не окнам, а программам (при запуске приложения закрепленная за ним кнопка в панели задач подсвечивается) и их можно переставлять в желаемом порядке.

Панель задач не содержит названий программ, а все активные приложения обозначаются одними пиктограммами. Однако текст может быть включен из меню Свойства панели задач, но в этом случае он будет отображаться только для активных программ, которые развернуты на рабочем столе.

Перемещение курсора по иконкам панели задач вызывает появление эскизов (миниатюрных версий) окон открытых приложе-

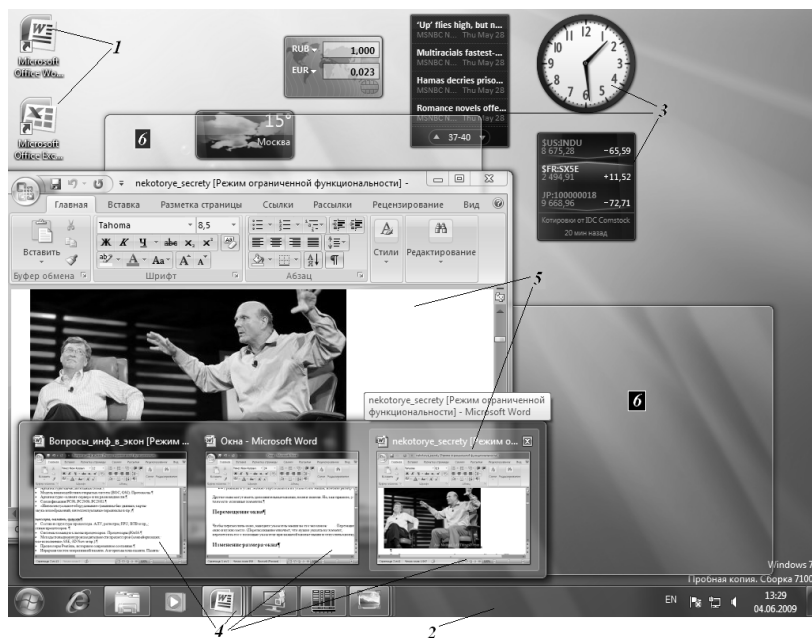


Рис. 2.43. Интерфейс Windows 7 (RC):

1 — значки рабочего стола; 2 — панель задач (TaskBar); 3 — мини-приложения (Gadgets) — погода, новости, курсы валют; 4 — иконка в панели задач и эскизы окон открытых файлов выбранного активного приложения (MS Word); 5 — окно выбранного файла и его эскиз; 6 — «прозрачные окна» невыбранных активных приложений

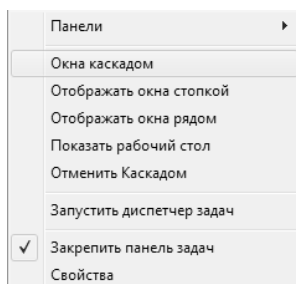
ний, которые достаточно велики для того, чтобы разглядеть содержимое, к тому же они обновляются в режиме реального времени.

Если приложение работает с несколькими документами (файлами), на рис. 2.43, 4, 5 в Word открыто 3 файла, это отображается на панели задач в виде «стопки ярлычков» и несколькими эскизами окон (три на рис. 2.43, 4). При наведении, в свою очередь, курсора на один из эскизов окон соответствующее окно открывается на переднем плане рабочего стола операционной системы, при этом другие окна становятся практически прозрачными и предстают в виде стеклянных рамок (рис. 2.43, 5, 6). Подобный подход значительно упрощает управление окнами, особенно если их открыто десять или более.

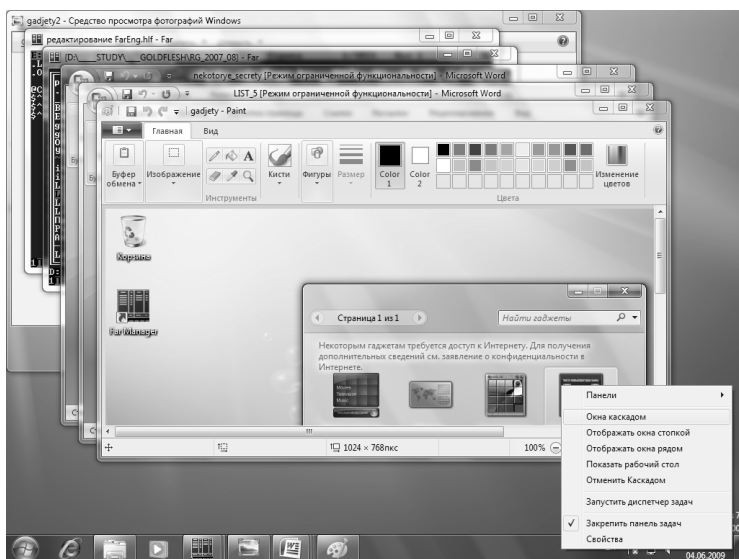
В связи с неудобством использования боковой панели Windows Vista, в которой находились мини-приложения (мини-программы для быстрого доступа к различной как систем-

ной, так и регулярно обновляемой через Internet информации), в Windows 7 их ярлыки могут произвольно перемещаться по рабочему столу посредством «буксировки» (рис. 2.43, 3).

Управление окнами приложений. При щелчке правой кнопки мыши по панели задач появляется меню отображения (рис. 2.44, а), где, в частности, можно установить вид размещения окон приложений на рабочем столе (рис. 2.44, б—г).



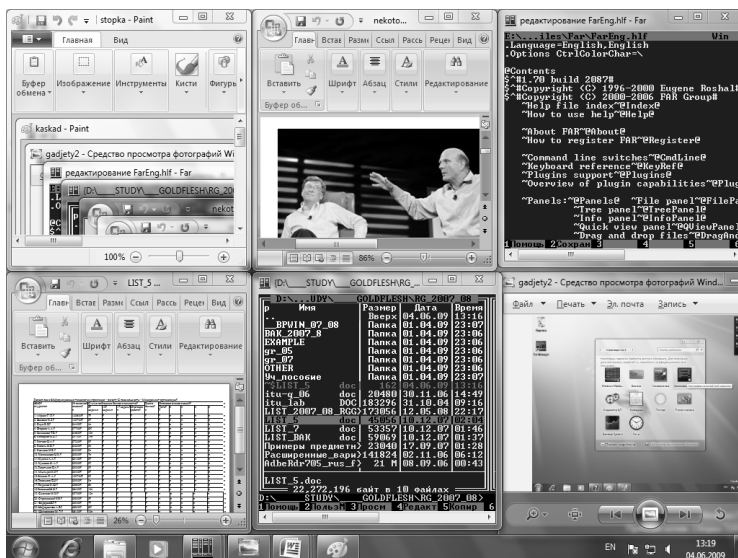
а



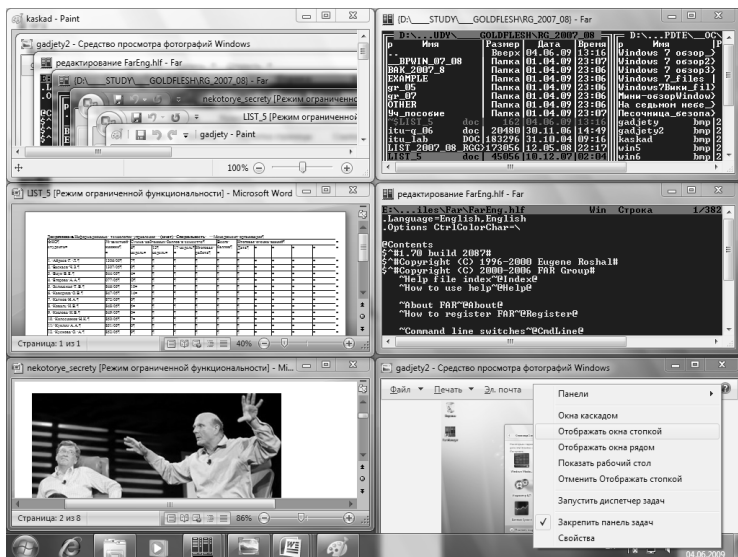
б

Рис. 2.44. Управление размещением окон (начало):

а — меню управления; б — размещение каскадом; в — размещение рядом; г — размещение стопкой



6



2

Рис. 2.44. Управление размещением окон (окончание)

Кроме того, к новым возможностям интерфейса можно отнести функцию Aero Shake — если захватить мышью окно и немного «потрясти» его, то все остальные открытые окна на рабочем столе свернутся в панель задач. Если окно «потрясти» вновь, то все окна возвратятся на свои места.

В Windows 7, если подтащить окно программы к верху экрана, оно автоматически распахнется на весь экран, а если оттащить — возвратится в исходное состояние. Если же переместить его в правый или левый угол, оно займет ровно половину экрана. Наконец, если отвести курсор в правый нижний угол, то от окон останутся только полупрозрачные рамки, и взору пользователя предстанет рабочий стол Windows с ярлыками приложений.

Модификация приложений. Остановимся на двух хорошо знакомых приложениях — редакторов текста WordPad и графики Paint (рис. 2.45). Они получили радикально переделанный интерфейс в стиле Office 2007 (Scenic Ribbon — Лента).

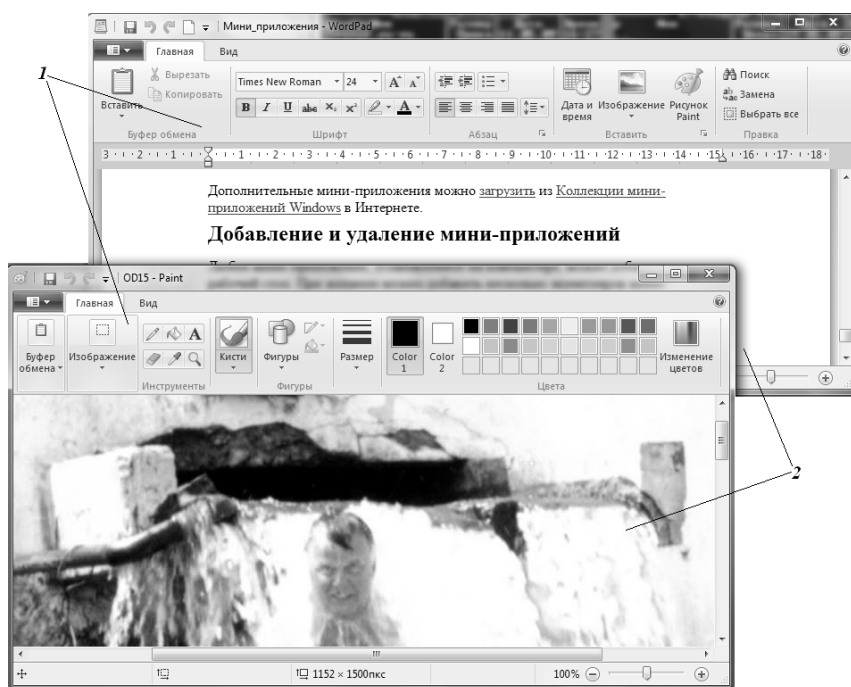


Рис. 2.45. Окна приложений WordPad и Paint:
1 — интерфейс Ribbon (Лента); 2 — рабочее окно

Функционально WordPad получил возможность открывать и сохранять документы форматов Office Open XML (.docx) и OpenDocument Format (.odt). В новом редакторе появился механизм вставки изображений, добавились расширенные функции работы с текстом, опция быстрой отправки созданного файла по электронной почте и прочие ранее отсутствовавшие инструменты.

Paint тоже получил такую же панель инструментов, как и в Word 2007, а также Paint обогатился на десяток новых инструментов работы с графикой (естественные кисти, подобные акварельной кисти, карандашу и мелку и пр.). Органы управления функциями интерфейса Paint специально разработаны так, чтобы их можно было использовать в паре с сенсорным экраном, поэтому пользователь может выбирать фигуры или рисовать новыми кистями одним или двумя пальцами.

Функция одновременного ввода (технологии Multi-Touch). Мульти-тач — поддержка сенсорного управления с помощью нескольких пальцев (одновременный ввод) — позволяет управлять компьютером через сенсорный экран хоть десятью пальцами обеих рук, по-новому взаимодействовать с файлами и программами. Например, можно, «трогая руками», менять размеры фотографий, сортировать и др.

Кроме того, известно большое количество приложений, использующих мультитач: цифровое фортепиано, «детские» графические редакторы, где можно рисовать сразу десятью пальцами.

Если Microsoft докажет преимущество данной технологии производителям оборудования, следует ожидать серьезного увеличения спроса на рынке сенсорных дисплеев.

Файловые системы NTFS4 (Windows NT) и NTFS5 (Windows 2000)

NTFS была разработана из файловой системы HPFS, разрабатывавшейся совместно IBM и Microsoft для проекта OS/2. Она начала использоваться вместе с Windows NT 3.1 в 1993 г. Windows NT 3.1 должна была составить конкуренцию серверам на базе NetWare и Unix, поэтому NTFS вобрала в себя все тогдашние технологические достижения. Рассмотрим файловую систему NTFS.

Структура файловой системы. Как и любая другая система, NTFS делит все полезное пространство диска на кластеры — блоки данных, используемые одновременно. NTFS поддер-

живает различные размеры кластеров — от 512 байт до 64 Кбайт, стандартом считается кластер размером 4 Кбайт.

Диск NTFS условно делится на две части (рис. 2.46). Первые 12 % диска отводятся под так называемую MFT-зону — пространство, в котором размещен метафайл MFT (Master File Table). Запись каких-либо данных в эту область невозможна. MFT-зона всегда держится пустой — это делается для того, чтобы главный служебный файл (MFT) не фрагментировался при своем расширении. Остальные 88 % диска представляют собой пространство для размещения файлов.

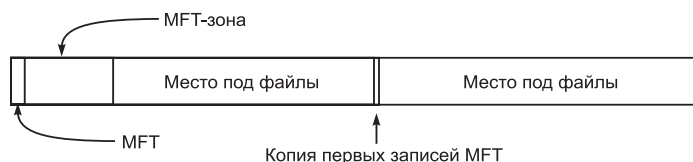


Рис. 2.46. Структура диска NTFS

Свободное место диска, однако, включает в себя все физически свободное место (незаполненные участки MFT-зоны туда тоже включаются). Механизм использования MFT-зоны таков: когда файлы уже нельзя записывать в обычное пространство, MFT-зона сокращается, освобождая место для записи файлов. При освобождении участка обычной области MFT-зона может снова расширяться.

Структура MFT. Каждый элемент файловой системы NTFS представляет собой файл — даже служебная информация. Как уже говорилось, главный файл NTFS называется MFT, или Master File Table — главная таблица файлов, которая размещается в MFT-зоне и представляет собой централизованный каталог всех остальных файлов диска. MFT поделен на записи фиксированного размера (обычно 1 Кбайт), и каждая запись соответствует какому-либо файлу. Первые 16 файлов носят служебный характер и недоступны операционной системе; они называются метаяйлами, причем самый первый из них — сам MFT. Эти первых 16 элементов MFT — единственная часть диска, имеющая фиксированное положение. Остальная часть MFT-файла может располагаться, как и любой другой файл, в произвольных местах диска; восстановить его положение можно с помощью него самого, используя за основу первый элемент MFT.

Все пространство тома NTFS представляет собой либо файл, либо часть файла. Главная таблица файлов содержит, по крайней мере, одну запись для каждого файла тома, включая одну запись для самой себя.

Все файлы на томе NTFS идентифицируются номером файла, который определяется позицией файла в MFT. Каждый файл и каталог на томе NTFS определяются набором атрибутов.

Базовая единица распределения дискового пространства для файловой системы NTFS — кластер. Размер кластера выражается в байтах и всегда равен целому количеству физических секторов. В качестве адреса файла NTFS использует номер кластера, а не физическое смещение в секторах или байтах.

Загрузочный сектор тома NTFS располагается в начале тома, а его копия — в середине тома. Загрузочный сектор состоит из стандартного блока параметров BIOS, количества секторов в томе, а также начального логического номера кластера основной копии MFT и зеркальной копии MFT.

Каждый *атрибут* файла NTFS состоит из полей: тип атрибута, длина атрибута, значение атрибута и, возможно, имя атрибута.

Имеется *системный набор атрибутов*, определяемых структурой тома NTFS. Системные атрибуты имеют фиксированные имена и коды их типа, а также определенный формат. Могут применяться также атрибуты, определяемые пользователями. Их имена, типы и форматы задаются исключительно пользователем. Атрибуты файлов упорядочены по убыванию кода атрибута, причем атрибут одного и того же типа может повторяться несколько раз. Существует два способа хранения атрибутов файла — резидентное хранение в записях таблицы MFT и нерезидентное хранение вне ее. Сортировка может осуществляться только по резидентным атрибутам. Файлы NTFS состоят, по крайней мере, из атрибутов, приведенных в табл. 2.9.

Размещение файлов. *Небольшие файлы (small).* Если файл имеет небольшой размер, то он может целиком располагаться внутри одной записи MFT размером 2 Кбайт (рис. 2.47, а). Из-за того, что файл может иметь переменное количество атрибутов, а также из-за переменного размера атрибутов нельзя наверняка утверждать, что файл уместится внутри записи. Однако обычно файлы размером менее 1500 байт помещаются внутри записи MFT.

Большие файлы (Large). Если файл не вмещается в одну запись MFT, то этот факт отображается в значении атрибу-

Таблица 2.9. Список атрибутов файла (на примере ФС NTFS)

№	Атрибут	Пояснение
1	Attribute List	Определяет список атрибутов, которые являются допустимыми для данного конкретного файла
2	File Name (FN)	Содержит длинное имя файла, а также номер входа в таблице MFT для родительского каталога; если этот файл содержится в нескольких каталогах, то у него будет несколько атрибутов типа File Name; этот атрибут всегда должен быть резидентным
3	MS-DOS Name	Имя файла в формате 8.3
4	Version	Атрибут содержит номер последней версии файла
5	Security Descriptor (SD)	Информация о защите файла: список прав доступа ACL и поле аудита, которое определяет, какого рода операции над этим файлом нужно регистрировать
6	Volume Version	Версия тома, используется только в системных файлах тома
7	Volume Name	Метка тома
8	Volume Information	Номер версии NTFS
9	Data	Содержит обычные данные файла
10	MFT bitmap	Содержит карту использования секторов на томе
11	Index Root	Корень В-дерева, используемого для поиска файлов в каталоге
12	Index Allocation	Нерезидентные части индексного списка В-дерева
13	External Attribute Information	Номер первого кластера и количество кластеров нерезидентного атрибута
14	Standard Information (SI)	Хранит всю остальную стандартную информацию о файле, которую трудно связать с каким-либо из других атрибутов файла, например, время создания файла, время обновления и др.

та «данные», который содержит признак того, что файл является нерезидентным и находится вне таблицы MFT. В этом случае атрибут «данные» содержит номер кластера для первого кластера каждого фрагмента данных (data run), а также количество непрерывных кластеров в каждом фрагменте (рис. 2.47, б).

Очень большие файлы (huge). Если файл настолько велик, что его атрибут данных не помещается в одной записи, то этот атрибут становится нерезидентным, т. е. он размещается в

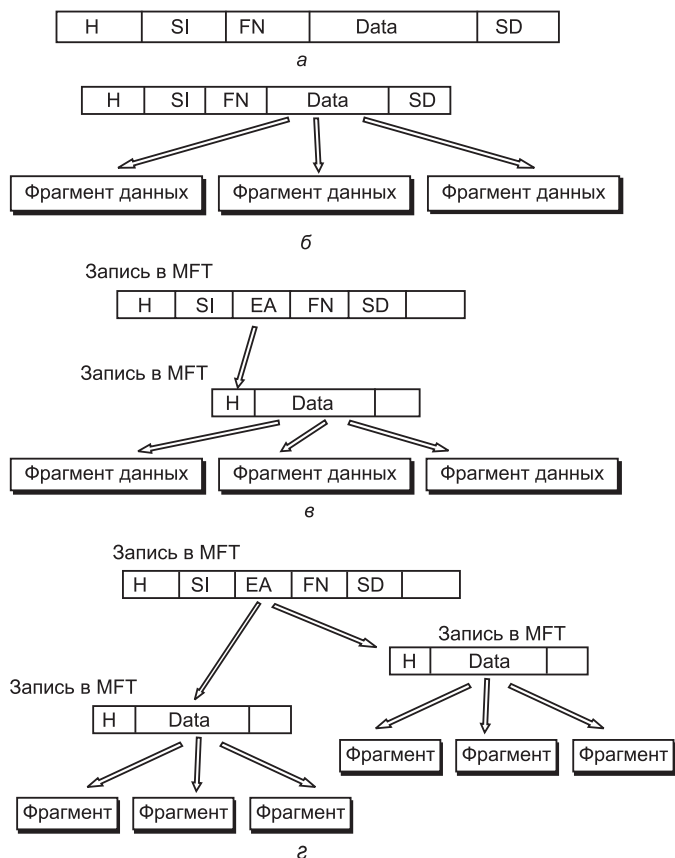


Рис. 2.47. Размещение файлов в NTFS:

a — небольшие файлы: H — заголовок (header); SI — стандартная информация (standard information); FN — имя файла (file name); Data — данные; SD — дескриптор безопасности (security descriptor); *б* — большие файлы; *в* — очень большие файлы: EA — внешний атрибут (external attribute); *г* — сверхбольшие файлы

другой записи таблицы MFT, ссылка на которую помещена в исходной записи о файле (рис. 2.47, в). Эта ссылка называется внешним атрибутом (external attribute). Нерезидентный атрибут содержит указатели на фрагменты данных.

Сверхбольшие файлы (extremely huge). Для сверхбольших файлов внешний атрибут может указывать на несколько нерезидентных атрибутов (рис. 2.47, г). Кроме того, внешний атрибут, как и любой другой атрибут, может храниться в нерези-

дентной форме, поэтому в NTFS не может быть атрибутов слишком большой длины, которые система не может обработать.

Каталоги. Каждый каталог NTFS представляет собой один вход в таблицу MFT, который содержит список файлов специальной формы, называемый индексом (index). Индексы позволяют сортировать файлы для ускорения поиска, основанного на значении определенного атрибута. В файловых системах FAT и HPFS используется сортировка файлов по имени. NTFS позволяет использовать для сортировки любой атрибут, если он хранится в резидентной форме. Имеется две формы списка файлов.

Небольшие списки файлов (small indexes). Если количество файлов в каталоге невелико, то список файлов может быть резидентным в записи в MFT, являющейся каталогом. В этом случае он называется небольшим каталогом (рис. 2.48, а). Небольшой список файлов содержит значения атрибутов файла. По умолчанию — это имя файла, а также номер записи MFT, содержащей начальную запись файла.

Большие списки файлов (large index). По мере того, как каталог растет, список файлов может потребовать нерезидентной формы хранения. Однако начальная часть списка всегда остается резидентной в корневой записи каталога в таблице MFT (рис. 2.48, б). Имена файлов резидентной части списка файлов являются узлами В-дерева. Остальные части списка файлов раз-

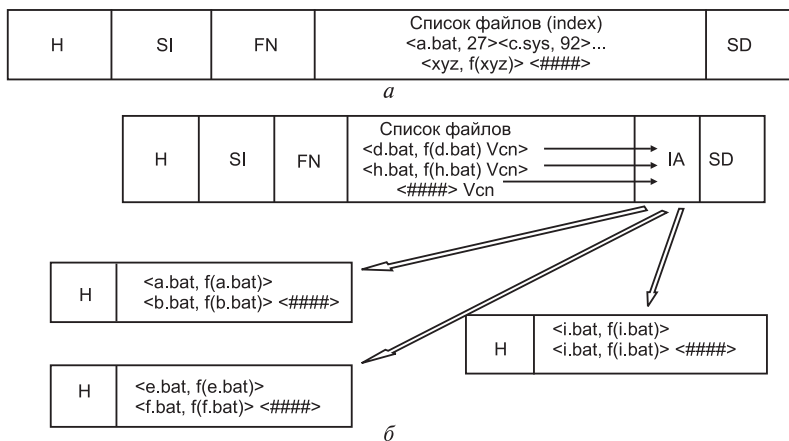


Рис. 2.48. Каталоги в NTFS:

а — небольшие каталоги (#### — признак конца списка файлов); б — большие каталоги

мешаются вне MFT. Для их поиска используется специальный атрибут «размещение списка» (Index Allocation — IA), представляющий собой набор номеров кластеров, которые указывают на остальные части списка. Одни части списков являются *листьями дерева*, а другие — *промежуточными узлами*, т. е. содержат наряду с именами файлов атрибут Index Allocation, указывающий на списки файлов более низких уровней.

Имена файлов. NTFS поддерживает имена файлов длиной до 255 символов. Имена файлов NTFS используют набор UNICODE с 16-битовыми символами. NTFS автоматически генерирует поддерживаемое MS DOS имя для каждого файла. Таким образом, файлы NTFS могут использоваться в сети операционными системами MS DOS и OS/2.

Поскольку NTFS использует набор символов UNICODE для имен файлов, существует возможность использования некоторых запрещенных в MS DOS символов. Для генерации короткого имени файла в стиле MS DOS NTFS удаляет все запрещенные символы, точки (кроме одной), а также любые пробелы из длинного имени файла. Далее имя файла усекается до 6 символов, добавляется тильда (~) и номер. Расширение имени файла усекается до 3 символов.

Короткие имена файлов с длинными именами на кириллице образуются по особой схеме, в зависимости от типа используемой файловой системы.

Надежность NTFS. NTFS является восстанавливаемой (recoverable) файловой системой, которая может привести себя в корректное состояние при практически любых реальных сбоях. Любая современная файловая система основана на таком понятии, как транзакция — действие, совершаемое целиком и корректно или не совершаемое вообще.

Журналирование — средство, позволяющее существенно сократить число ошибок и сбоев системы. Опыт показывает, что NTFS восстанавливается в полностью корректное состояние даже при сбоях в очень загруженные дисковой активностью моменты. Она гарантирует согласованность данных тома, используя стандартную процедуру регистрации транзакций. Каждая операция ввода-вывода, которая изменяет файл на томе NTFS, рассматривается файловой системой как транзакция.

При модификации файла специальная компонента файловой системы — сервис регистрации файлов (Log File Service) — фиксирует всю информацию, необходимую для по-

вторения (redo) или отката (undo) транзакции в специальном файле с именем \$LogFile. Если транзакция не завершается нормально, то NTFS пытается закончить транзакцию (повторить) или производит ее откат.

Динамические диски (Dynamic Disk) — физический диск, на котором могут быть созданы динамические разделы. Динамические разделы могут быть следующих видов:

- простые (simple) — практически ни чем не отличаются от обычных;
- составные (spanned) — состоят из нескольких динамических дисков, которые представлены как один диск. Данные пишутся и читаются последовательно;
- чередующие (striped) — несколько динамических дисков, которые представлены как один диск. Данные пишутся и читаются одновременно на несколько дисков. Это, теоретически, должно обеспечивать вдвое большую скорость на дисковых операциях. На практике, прирост хотя и значительный, но меньше чем в 2 раза;
- зеркальные (mirrored) — состоят из двух физических дисков. Данные, записываемые на один из дисков, автоматически дублируются на другом. Это не дает никаких преимуществ в плане скорости, но зато обеспечивает вдвое большую степень надежности сохранности данных.

Обычный диск может быть конвертирован в динамический из окна Disk Management, однако обратный процесс (конвертировать динамический диск в простой) не всегда возможен. Например, если диск с самого начала создавался как динамический, то на нем отсутствует привычная таблица разделов, и чтобы создать ее, его придется заново разбивать с помощью fdisk и форматировать.

RAID. Для обеспечения сохранности пользовательских данных используется программная поддержка массивов RAID (Redundant Array of Inexpensive/Independent Disks). В сочетании с поддержкой зеркализации дисков или расщепления с контролем четности (RAID 5) NTFS может выдержать любой одиночный сбой. В Windows NT поддерживаются уровни 0, 1 и 5. В RAID 0 данные расщепляются на блоки по 64 Кбайт, поддерживается от 2 до 32 дисков. RAID 1 осуществляется на уровне разделов, т. е. зеркализируются именно разделы. При отказе зеркализованного раздела администратор должен отменить отношения зеркализации, чтобы использовать оставшийся раздел как отдельный том. Затем можно использовать

свободный раздел на другом диске, чтобы вновь установить зеркальные отношения. Зеркализации может быть подвергнут любой раздел, включая загрузочный (Boot Partition). В принципе зеркализация является более дорогим способом, чем другие, так как коэффициент использования дискового пространства составляет только 50 %, с другой стороны, для небольших сетей это весьма приемлемый вариант, так как для его реализации достаточно только двух дисков.

RAID 5 требует минимум трех дисков (максимум 32 диска), поддерживает файловые системы FAT, NTFS, причем загрузочный раздел не может быть расщеплен. Если отказывает диск, входящий в состав массива RAID 5, то компьютер может продолжать работу и получать доступ к данным. Однако данные отказавшего диска будут в течение всего времени регенерироваться на основании данных других дисков, и производительность системы может упасть. Можно воссоздать данные отказавшего диска на новом диске. Для этого нужно иметь свободный раздел на каком-либо работоспособном диске равного или большего размера, чем отказавший. Затем запускается процедура восстановления данных из пункта Regenerate меню Fault Tolerance утилиты Disk Manager.

NTFS поддерживает также «горячее» переназначение секторов, когда при возникновении ошибки из-за наличия плохого сектора данные переписываются в новый хороший сектор, а сбойный исключается из работы. Администратор уведомляется с помощью утилиты просмотра событий Event Viewer о всех событиях, связанных с обработкой сбойных секторов, а также о потенциальной угрозе потери данных, если избыточная копия также отказывает.

Сжатие. NTFS имеет встроенную поддержку сжатия дисков — то, для чего раньше приходилось использовать Stacker или DoubleSpace. Любой файл или каталог в индивидуальном порядке может храниться на диске в сжатом виде — этот процесс прозрачен для приложений. Сжатие файлов осуществляется с высокой скоростью, однако при этом часто возникает отрицательный эффект — фрагментация сжатых файлов. Сжатие осуществляется блоками по 16 кластеров и использует так называемые виртуальные кластеры — гибкое решение, позволяющее добиться полезных эффектов, например, половина файла может быть сжата, а половина — нет. Это достигается благодаря тому, что хранение информации о компрессирован-

ности определенных фрагментов очень похоже на обычную фрагментацию файлов.

Сжатый файл имеет «виртуальные» кластеры, реальной информации в которых нет. Как только система обнаруживает такие виртуальные кластеры, это означает, что данные предыдущего блока, кратного 16 кластерам, должны быть разжаты, а получившиеся данные должны заполнить виртуальные кластеры (рис. 2.49).

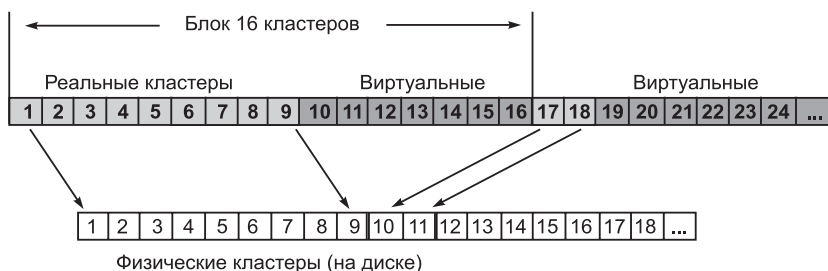


Рис. 2.49. Реализация сжатия данных

Hard Link — один и тот же файл может иметь два имени (несколько указателей файла-каталога или разных каталогов ссылаются на одну и ту же MFT-запись). Допустим, один и тот же файл имеет имена 1.txt и 2.txt, и если пользователь удалит файл 1.txt, останется файл 2.txt, наоборот, если сотрет 2.txt — останется файл 1.txt, т. е. оба имени с момента создания файла равноправны. Файл физически удаляется лишь тогда, когда будет удалено его последнее имя.

Шифрование (NT5). Каждый файл или каталог может также быть зашифрован, что не даст возможности прочесть его другой инсталляцией ОС NT. В сочетании со стандартным паролем на загрузку системы, эта возможность обеспечивает достаточную для большинства применений безопасность избранных пользователем важных данных.

В табл. 2.10 приведены некоторые характеристики ФС ряда различных ОС.

Файловая система NTFS5. Кроме непосредственного изменения самой структуры NTFS, в W2k добавлен Microsoft Index Server, который значительно ускоряет поиск файлов, особенно по их содержимому, за счет индексации содержимого дисков. Управляется эта служба через раздел Служба индексирования окна Управление компьютером. В этом разделе можно просмотреть,

Таблица 2.10. Сравнительные характеристики ФС MS Windows

Характеристика	Файловая система		
	FAT16	FAT32	NTFS
Системы, ее поддерживающие	DOS, Windows9X, NT всех версий	Windows98, NT5, XP	NT4, NT5, XP
Максимальный размер тома	2 Гбайт	Практически не ограничен	Практически не ограничен
Максимальное число файлов на томе	Примерно 65 тысяч	Практически не ограничено	Практически не ограничено
Имя файла	С поддержкой длинных имен — 255 символов, системный набор символов	С поддержкой длинных имен — 255 символов, системный набор символов	255 символов, любые символы любых алфавитов (65 тысяч разных начертаний)
Возможные атрибуты файла	Базовый набор	Базовый набор	Любая информация, актуальная с точки зрения производителей программного обеспечения
Безопасность	Нет	Нет	Начиная с NT 5.0 встроена возможность физически шифровать данные
Сжатие	Нет	Нет	Да
Устойчивость к сбоям	Средняя (система слишком проста и поэтому мало что подвержено разрушению)	Плохая (средства оптимизации по скорости привели к появлению слабых по надежности мест)	Полная — автоматическое восстановление системы при любых сбоях (не считая физические ошибки записи, когда пишется одно, а на самом деле записывается другое)
Экономичность	Минимальная (велики размеры кластеров на больших дисках)	Улучшена за счет уменьшения размеров кластеров	Максимальная. Очень эффективная и разнообразная система хранения данных
Быстродействие	Высокое для малого числа файлов, но быстро уменьшается с появлением большого количества файлов в каталогах; результат — для слабо заполненных дисков — максимальное, для заполненных — плохое	Полностью аналогично FAT, но на дисках большого размера (десятки гигабайт) начинаются серьезные проблемы с общей организацией данных	Система не очень эффективна для малых и простых разделов (до 1 Гбайт), но работа с огромными массивами данных и внушительными каталогами организована как нельзя более эффективно и очень сильно превосходит по скорости другие системы

какие директории индексируются, и, при желании, добавить новые или удалить старые (служба работает с любыми разделами, а не только NTFS).

Второе, достаточно важное отличие NTFS5 от старой версии — возможность поиска файла по имени его владельца. Если нужно найти все файлы, созданные в Word каким-то одним пользователем, на диске, где этих файлов тысячи, то в NT4 это было проблемой. С помощью Access Control List (Список управления доступом) можно легко проверить, какие файлы доступны пользователю, и установить права доступа к отдельным файлам или каталогам.

В NTFS5 добавлена функция точки монтирования или, иначе говоря, точки соединения (junction point). Функция эта знакома пользователям различных версий Unix/Linux (см. рис. 3.41), но в продуктах Microsoft она здесь появляется впервые.

2.6. Операционные системы других аппаратных платформ

Рассмотрим вкратце в заключение главы примеры некоторых операционных систем, обслуживающих персональные компьютеры и прочие устройства других (не IBM PC-совместимых) платформ, а именно:

- Windows Mobile (Microsoft) — ОС для карманных ПК (КПК), смартфонов и коммуникаторов;
- Android (Google) — портативная ОС для коммуникаторов, КПК, планшетных компьютеров, электронных книжек, цифровых проигрывателей, наручных часов, нетбуков, базирующуюся на ядре Linux;
- Mac OS (Apple Computer) — семейство систем для ПК Macintosh.

Windows Mobile (WM)



WM — компактная ОС, разработанная Microsoft, и ориентированная на мобильные устройства (mobile devices). Она базируется на Windows CE и спроектирована так, что интерфейс аналогичен версиям Windows для настольных ПК.

ОС WM предназначена для эксплуатации на двух типах мобильных устройств:

- Pocket PC (тип КПК — карманного персонального компьютера, или personal digital assistant — PDA);
- Windows Phone (тип смартфонов).

Основным устройством, поддерживающим WM считается КПК, который не предназначен для выполнения функций мобильного телефонного аппарата (или коммуникатор). Такое устройство (Windows Mobile Classic devices) ранее было известно как «Pocket PC». Современные версии WM, предназначенные для совместимых с Pocket PC устройств, официально называются «Windows Mobile 6 Professional» (для систем с возможностями мобильного телефона) и «Windows Mobile 6 Classic» (для систем без таких возможностей).

Windows Phone — следующий тип устройств, которые поддерживаются WM, начиная с версии 2002 (Pocket PC phone 2002). В обычном понимании «смартфон» означает мобильный телефон с функциями КПК, однако фирмой Microsoft термин «Smartphone» используется для обозначения устройств, которые в отличие от «Windows Phone» ориентированы на работу с помощью одной руки, обладают меньшим разрешением экрана и в которых отсутствует сенсорный экран.

Windows Mobile для Pocket PC поддерживает следующие стандартные возможности для большинства версий:

- экран Сегодня (Today) — рис. 2.50, а, включающий текущую дату, информацию пользователя, сообщения электронной почты и задачи;
- панель задач (taskbar), показывающая текущее время, наличие GPRS-трафика, состояние приема сотовой связи, уведомления о звонках;
- главное меню Пуск — рис. 2.50, б, которое содержит список приложений, в том числе:
 - мобильный офис (Office Mobile) — версия MS Office для КПК (рис. 2.50, в);
 - почтовый клиент Outlook Mobile;
 - Internet-браузер Explorer Mobile (рис. 2.50, г);
 - мультимедийный проигрыватель Media Player, т. д.;
- файловая система FAT16, совместимая с Windows 9x/NT;
- синхронизация содержимого пользовательских папок с настольным ПК через интерфейсы Bluetooth или USB (клиент-серверное приложение ActiveSync, рис. 2.51);
- поддержка многозадачности.

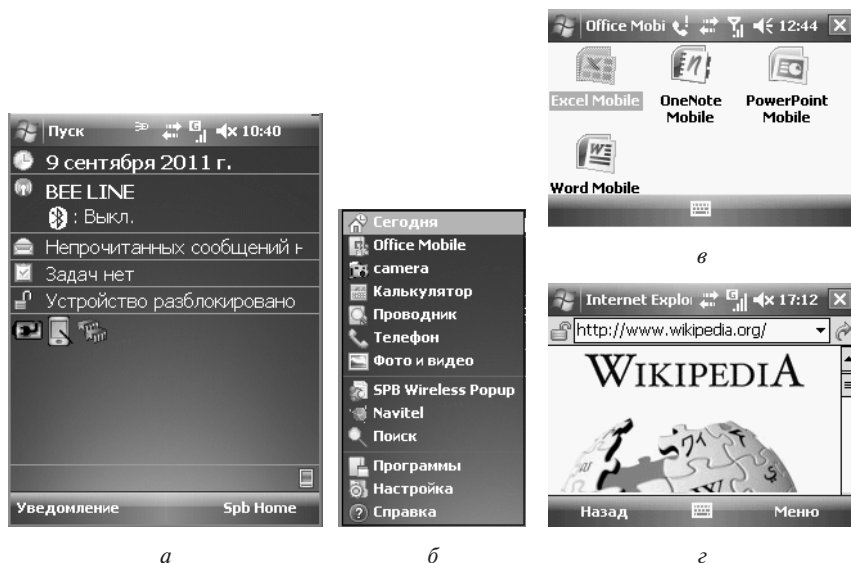


Рис. 2.50. Элементы интерфейса Windows Mobile 6:

а — экран Сегодня; *б* — меню Пуск; *в* — компоненты Office Mobile; *г* — экран Explorer Mobile

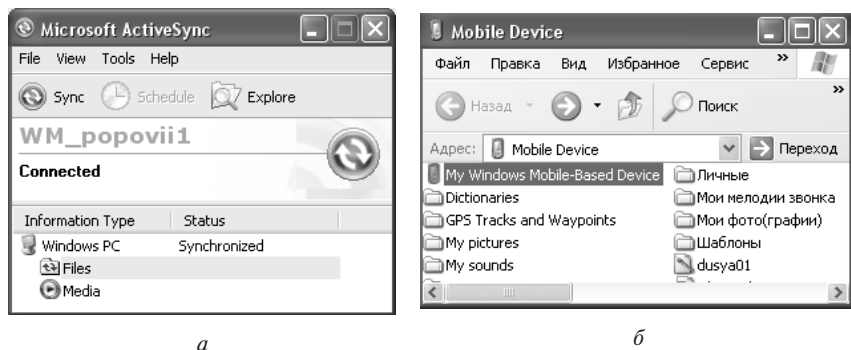


Рис. 2.51. Окна приложения ActiveSync (на экране ПК, ОС WinXP):

а — окно соединения; *б* — окно просмотра содержимого памяти мобильного устройства

Microsoft WM не является единственным изделием подобного назначения. Широко распространены также ОС Nokia Symbian OS, PalmOS, Apple iPhone OS, BlackBerry OS и ряд систем на базе ядра Linux, например Google Android, Palm WebOS, Access Linux

Platform и др. Некоторые производители мобильных устройств выпускают собственные ОС (например, Samsung bada).

Рассмотрим вкратце некоторые интересные приложения для ОС WM (и других мобильных ОС).



Оболочка ОС WM Spb Mobile Shell позволяет видоизменить пользовательский интерфейс коммуникатора, сделать его более удобным для управления, сократив до минимума количество операций со стилусом. Утилита оптимизирована под управление пальцами, при этом все функции КПК на платформе Windows Mobile сохраняются. Обладая широкими возможностями для персонализации, программа отображает всю важную информацию на одном экране. Доступны создание закладок, способных показывать автоматически обновляемый прогноз погоды, будильники и мировое время (рис. 2.52, а); а также быстрый набор по фотографии абонента (рис. 2.53, б). Кнопка Запуск обеспечивает доступ к наиболее востребованным приложениям (рис. 2.53, в). Пункт меню Настройки позволяет регулировать количество закладок на дисплее и устанавливать дополнительные темы, основанные на оттенках цветов.



Рис. 2.52. Некоторые экраны SPB Shell:

а — Окно SPB Home; б — экран Контакты; в — окно Запуск



My Mobiler (разработка MTUX) — программный продукт, обеспечивающий простой и прозрачный для пользователя обмен данными между КПК и персональным компьютером (ПК).

Он позволяет манипулировать мобильным устройством с персонального компьютера: заносить данные в КПК при помощи стандартной клавиатуры или мыши, производить операции копирования/вставки между двумя устройствами, делать скриншоты с экрана КПК, обмениваться файлами (рис. 2.53, а).



Рис. 2.53. Окно приложения на экране ПК, ОС WinXp: а — MyMobiler; система Navitel: б — Карта; в — Спутники

Для работы MTUX My Mobiler требует соединения КПК и ПК через ActiveSync или Windows Mobile Device Center.

Навител Навигатор — навигационная система для коммуникаторов и КПК, снабженных ГЛОНАСС/GPS-приемником. Программа предназначена не только для автолюбителей, но и для пеших прогулок с навигатором по городу, либо по пересеченной местности. В навигационной системе используются подробные векторные карты городов и областей России, с названиями улиц, номерами домов, станциями метро, АЗС, ресторанами, расположением лесов, рек, озер и другой важной и полезной информацией, облегчающей ориентирование как в мегаполисе, так и за его пределами. Навител Навигатор позволяет использовать карты, созданные и обновляемые самими пользователями, в популярном картографическом редакторе GPSMapEdit (рис. 2.53, б, в).



Первая версия ОС разрабатывалась компанией Android Inc., которую в 2005 г. приобрела Google. Впоследствии Google инициировала (5.09.2007 г.) создание альянса Open Handset Alliance (ОНА), целью которого является разработка открытых стандартов для мобильных устройств. В настоящий момент ОНА объединяет 34 компании, среди которых крупнейшие сотовые операторы, производители мобильных устройств, разработчики микросхем. Одновременно с представлением ОНА была анонсирована открытая мобильная платформа Android, основанная на ядре Linux.

Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit создаёт приложения, написанные на Си и других языках.

По утверждению специалистов, в 75 % смартфонов, проданных в третьем квартале 2012 г., была установлена операционная ОС Android.

К ноябрю 2012 г. выпущено 14 версий системы. Последняя версия — 4.2 Jelly Bean.

Приложения Android. 22.10.2008 г. Google объявила об открытии онлайн-магазина приложений для ОС Android — «Android Market». По соглашению разработчики получают 70 % прибыли,



Рис. 2.54. Некоторые экраны интерфейса ОС Android:
а — Приложения; б — Телефон; в — Настройки телефона

операторы сотовой связи — 30 %. В феврале 2009 г. для разработчиков из США и Великобритании появилась возможность брать плату за свои приложения в «Android Market». На декабрь 2011 г. с момента создания «Android Market» пользователями было скачано 10 млрд приложений.

В марте 2012 г. компания Google объединила мультимедийные сервисы «Книги», «Android Market», «Музыка» и другие в «Google Play». Интернет-магазин «Google Play» работает в 190 странах, насчитывает более 700 тысяч приложений, а за время работы сервиса было скачано 25 млрд приложений.

На рис. 2.54 приведены примеры экранов интерфейса ОС Android (Приложения, Телефон, Настройки телефона).

MacOS X



В качестве иллюстрации рассматривается версия Mac OS X 10.6.5 («Snow Leopard»).

Mac OS (Macintosh Operating System) представляет собой семейство операционных систем с графическим интерфейсом, ориентированных исключительно на компьютеры Apple Macintosh. Считается, что Mac OS была первой из современных ОС, в которой был реализован графический интерфейс вместо традиционной текстовой командной строки. Таким образом, Mac OS задала некую планку, к которой начали стремиться разработчики и других операционных систем. Кроме того, считается, что впоследствии ряд компаний использовали в своих продуктах идеи Apple, например, Microsoft, которая разработала подобную Mac OS графическую оболочку для своей операционной системы MS-DOS (Windows, в переводе с английского — окна, см. выше, п. 2.2 и далее).

Система Mac OS базировалась на прототипе графического интерфейса пользователя, заимствованного руководством Apple у исследовательского центра Xerox PARC. Разработчики Macintosh использовали некоторые идеи прототипа Xerox, расширив их и добавив собственные.

Краткая история системы.

System (24.01.1984). Первая операционная система Mac OS называлась «System 1.0», была однозадачной и в качестве файловой системы использовала MFS (Macintosh File System). Компьютеры Macintosh в это время базировались на процессорах Motorola 680x0.

System 7.5 (13.05.1991) стала последней из версий System.

Mac OS 8.1 (26.07.1998). Mac OS 8.1 стала одной из последних операционных систем Apple Computer, работающих на процессорах Motorola. В 1992 г. в результате выполнения совместного проекта фирм Apple, IBM и Motorola был разработан процессор PowerPC, который использовался в Apple Computer в дальнейшем более 10 лет.

С выходом Mac OS 8.1 компьютеры Apple Macintosh обзаводятся поддержкой USB и обновленной файловой системой HFS+. Появляются некоторые полезные технологии, например, печать на сетевом принтере. Поддержка файловой системы MFS была урезана до минимума — только чтение, позже в Mac OS 8.1 MFS не будет поддерживаться и уступит место более совершенной HFS.

Mac OS X 10.0 Cheetah (В данном случае X — это римская цифра «десять», поскольку первая официальная версия Mac OS X была одновременно десятой версией операционных систем для компьютеров Apple.) Первая Mac OS X 10.0 вышла 24.03.2001 и называлась «Cheetah» (Гепард). Главное отличие OS X от классической Mac OS состояло в том, что в основе системы лежали методология ОС Unix и технологии компании NeXT, которую Apple приобрела в 1997 г.

Mac OS X 10.1 Puma. В сентябре 2001 г. Apple выпускает Mac OS X 10.1 Puma, в которой основное внимание было уделено производительности и стабильности работы.

Mac OS X 10.2 Jaguar (23.08.2002). Здесь впервые появляются такие инструменты и технологии как Bonjour, iChat, Address Book, CUPS (Common Unix Printing System). Был обновлен Sherlock, повышена производительность системы, появилась журналируемая файловая система (начиная с 10.2.3).

Mac OS X 10.3 Panther (24.10.2003). Система оказалась настолько удачной, что некоторые пользователи и сегодня используют Panther для несложных задач вроде работы с текстом и серфинга Internet.

Mac OS X 10.4 Tiger (29.04.2005). Система получает много новых функций: Automator, Grapher, Dictionary, Quartz Composer и Dashboard. Обновлены Spotlight, iChat, Safari RSS, VoiceOver, QuickTime, Xcode и др. С учетом опыта разработки предыдущей ОС (Panther) Apple решает отказаться от PowerPC в пользу процессоров Intel. Mac OS X 10.4 Tiger стала переходным вариантом, так как работала как на процессорах PowerPC, так и на Intel.

Mac OS X 10.5 Leopard (26.10.2007). Система стала логическим продолжением Tiger и также работала как на процессорах PowerPC, так и Intel.

Mac OS X 10.6 Snow Leopard. Первые упоминания об OS X 10.6 появились еще в 2008 г., Но только в июне 2009 г. Apple показала новую ОС и огласила дату выпуска — сентябрь 2009.

Mac OS X строится на базе микроядра, которое называется Darwin (рис. 2.55). Внутри Darwin находится «ядро в ядре» — микроядро Mach (рис. 2.56), созданное на базе BSD (см. далее, п. 3.3, 3.4) и являющееся основой для ряда Unix-подобных (точнее — BSD Unix-подобных) систем. Mach поддерживает такие низкоуровневые функции управления ресурсами, как:

- управление единицами выполнения (нитеями, потоками);
- назначение ресурсов для процессов (в терминологии Mach — задач, task);

```
Read HFS+ file: [hd(0,2)/System/Library/CoreServices/SystemVersion.  
Loading Darwin 10.6  
Loading kernel rd(0,0)/Extra/legacy_kernel  
Read HFS+ file: [rd(0,1)/Extra/legacy_kernel] 4096 bytes.  
Read HFS+ file: [rd(0,1)/Extra/legacy_kernel] 7457805 bytes.  
LoadDrivers: Loading from [rd(0,0)/Extra/Extensions.mkext]  
Read HFS+ file: [rd(0,1)/Extra/Extensions.mkext] 4096 bytes.  
Read HFS+ file: [rd(0,1)/Extra/Extensions.mkext] 1138859 bytes.
```

Рис. 2.55. Фрагмент протокола загрузки ОС MacOS X 10.6.5 (ядро Darwin 10.6)

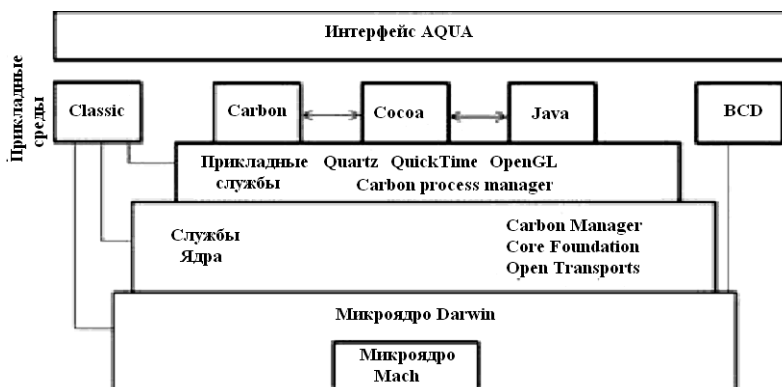


Рис. 2.56. Архитектура ОС Mac OS X

- поддержка адресных пространств задач;
- обмен сообщениями между задачами;
- управление реальными ресурсами (процессорами, памятью, вводом-выводом).

Общая архитектура Mac OS X иллюстрируется рис. 2.56.

Службы ядра содержат те системные сервисы, которые не связаны с графическим интерфейсом пользователя:

- Carbon Manager — общесистемные службы, обеспечивающие низкоуровневый сервис для всех прикладных сред, в том числе — обеспечение для приложения возможности находить во время выполнения различные программные объекты (компоненты), а также создавать компоненты; обеспечение абстрактных типов для коллекций данных; работа с датой, временем, географическими местами, временными зонами и т. п.; поддержка всех файловых систем; обеспечение работы с папками; управление виртуальной памятью; средства для создания потоков, управления ими и синхронизации;
- Core Foundation — каркас, который обеспечивает некоторые базовые программные службы сервисы, полезные для более высоких уровней программного обеспечения: набор инструментов для манипулирования строками, включая поддержку Unicode; средства организации и поиска различных типов программных ресурсов (исполняемых кодов, графических и звуковых образов и т. п.); обеспечение архитектуры подключений; средства доступа к локальным или удаленным ресурсам через URL; механизм обмена сообщениями (уведомлениями) между процессами;
- Open Transport — основные модули пользовательского уровня для обеспечения работы в сети и коммуникаций в Mac OS X.

Прикладные службы Mac OS X прежде всего предназначены для обеспечения графического и оконного интерфейса. Вот некоторые из них:

- Quartz — набор модулей, который состоит из двух частей: исполнения изображений (собственно Quartz) и базовых графических служб или сервера окон;
- QuickTime предоставляет средства для эффективной работы с мультимедийной информацией, такой как видеоролики, изображения, аудиозаписи. Компоненты QuickTime позволяют приложениям не зависеть в работе с мультиме-

дийной информацией от конкретных типов устройств и памяти;

- OpenGL — многоплатформенный промышленный стандарт для 3-мерного рисования и ускорения работы аппаратуры. Использование OpenGL обеспечивает высокую эффективность в создании анимации в реальном времени для игр и научной или деловой визуализации;
- Carbon Process Manager (CPM) обеспечивает абстракцию процесса для прикладных сред. В ядре процесс (задача) является сущностью, состоящей из набора потоков, адресного пространства и пространства имен портов. CPM на базе задачи ядра создает CPM-процессы, которые представляют процессы для прикладных сред.

Прикладные среды Mac OS X состоят из каркасов (framework), библиотек и сервисов, которые обеспечивают выполнение приложений в той или иной модели API. Mac OS X поддерживает следующие прикладные среды:

- Carbon — интерфейс прикладного программирования (API). Менеджеры Carbon выполняют обслуживание других прикладных сред;
- Cocoa — объектно-ориентированная среда для языков Java и Objective-C;
- Java позволяет разрабатывать и выполнять в Mac OS X приложения и апплеты, соответствующие спецификациям стандартной Java. Среда Java включает в себя компилятор javac и полный набор утилит, среду выполнения — виртуальную машину Java, базовый набор пакетов Java и пр.;
- Classic обеспечивает выполнение приложений Mac OS (предшествующей версии). В этой среде не обеспечиваются свойства, предоставляемые новым ядром ОС и интерфейсом Aqua. Эта среда не поддерживается прикладными службами непосредственно, следовательно, она обеспечивает только выполнение приложений, но не их разработку;
- BSD выполняет программы BSD из командной строки. Она обеспечивает shell и стандартный набор команд и утилит BSD. Эта среда базируется непосредственно на функциях ядра и не является обязательной для Mac OS X (может быть отменена при инсталляции).

Интерфейс Aqua. Наименование отражает тот факт, что при визуализации обычно используется метафора водного простран-

ства, отражающая такие свойства этой среды, как оттенки цвета, глубина, прозрачность, блики, течения и пр.

Среди элементов интерфейса Aqua можно назвать такие, как (рис. 2.57):

- Док (от Dock — акватория для размещения судов, с которыми осуществляются погрузочно-разгрузочные, ремонтные или иные работы) — функциональный аналог Линейки программ в выше рассмотренных системах;
- «Этикетки» (sheet) — диалоги, привязанные к соответствующему окну;
- иерархия окон, облегчающая ориентацию в многооконной среде;
- возможность масштабирования иконок от максимального размера 128 × 128 до мини-иконок;



Рис. 2.57. Элементы интерфейса AQUA:

1 — рисунок фона рабочего стола; 2 — панель Док; 3 — верхняя панель окна; 4 — верхнее меню (настройка навигатора Finder); 5 — меню настройки даты и времени; 6 — меню управления Dashboard; 7 — виджет Calculator; 8 — виджет World Clock; 9 — виджет Weather; 10 — окно Finder; 11 — стеки Дока (Программы, Документы, Загрузки)

- «выдвижные ящики» (drawer), содержащих те управляющие элементы окна, в постоянной визуализации которых нет необходимости;
- использование анимации для отображения изменения состояния элементов интерфейса;
- расширение возможностей использования клавиатуры.

Рассмотрим более подробно типичные элементы панели Dock:



Finder — навигатор и файловый менеджер. Используя встроенную технологию поиска Spotlight, приложение Finder становится мощным инструментом для поиска и организации файлов. Будучи встроенной функцией ядра Mac OS X, Spotlight производит поиск по всей системе, выдавая результаты поиска по всем приложениям и файлам. Она определяет местонахождение объектов по заданным пользователем критериям поиска с высокой точностью, даже при вводе только нескольких символов. Помимо функции моментального поиска по имени файла, Finder может находить документы по текстовому содержанию и метаданным. Результаты поиска можно сохранять в виде специальных интеллектуальных папок, которые динамически собирают вместе все файлы, соответствующие заданным критериям, независимо от того, где эти файлы реально находятся. Боковое меню в Finder упрощает доступ к папкам, размещенным на локальном компьютере или компьютерах, находящихся в общем доступе в сети, а также доступ к сохраненным вариантам поиска;



DashBoard — интерфейс прямого доступа к совокупности удобных и простых приложений, называемых виджетами. Вот некоторые из них: Stock (биржевые сводки), Calculator (калькулятор, рис. 2.57, 7), World Clock (время в различных часовых поясах, рис. 2.57, 8), Weather (погода — рис. 2.57, 9), Flight Tracker (информация об авиарейсах), Unit Converter (конвертер валют), Translator (переводчик), Calendar (календарь), iTunes (работа с музыкальными файлами), Tile Game (игры), Stickies (важные заметки) и пр.;



Mail — электронная почта. Пакет Mail обеспечивает новые возможности просмотра, поиска и создания интеллектуальных почтовых ящиков на основе поисковой технологии Spotlight, а также контроль доступа, механизм редактирования текста в формате HTML, синхронизацию параметров настройки почты в .Mac и возможности просмотра прикрепленных файлов с изображениями в режиме полноэкранной демонстрации;



Safari — мощный Web-браузер. В Safari применен стандарт RSS, что позволяет отображать заголовки последних новостей и краткое содержание статей ведущих информационных служб, таких как New York Times, CNN и BBC, в удобном для чтения и поиска формате в виде единого списка статей. Пользователь также может создать личную информационную службу новостей, делая закладки на темах, представляющих интерес; Safari будет автоматически обновлять результаты поиска по мере появления новых статей по отмеченным темам;



iChat — видео конференция, в которой могут принимать участие от одного до трех человек, в виртуальном месте встречи, с высококачественным звуком и высоким качеством изображения, соответствующим стандарту H-264;



AddressBook (адресная книга). Для встроенного менеджера контактов добавлены интеллектуальные группы, которые автоматически выбирают все контакты по заданным критериям. Кроме того, имеется функция печати конвертов, а также возможность совместного использования адресной книги членами семьи и друзьями с помощью приложения .Mac;



iCal представляет собой электронный календарь-планировщик, который помогает управлять задачами и планировать время более эффективно. С помощью пакета iCal можно объединять календари в группы, просматривать все уведомления о приглашениях в централизованном представлении, а также создавать отформатированную печатную копию календаря;



Time Machine — автоматически создает резервные копии данных, хранящихся на компьютере. При потере файла можно «переместиться во времени», чтобы восстановить его. Чтобы начать использование Time Machine, следует подключить внешний диск или флэш-накопитель и в появившемся диалоговом окне выбрать Использовать как резервный диск. При включении Time Machine программа создает резервную копию компьютера на указанном диске;



Automator — функция автоматизации выполнения сложных или повторяющихся задач без необходимости программирования. Приложение Automator, в котором используются простые операции типа «укажи и выбери» и «перетаскивания» с помощью манипулятора, упрощает создание и

совместное управление пользовательскими технологическими процессами;



Dictionary — словарь и тезаурус, основанные на «Новом оксфордском американском словаре» и «Словаре синонимов американской ассоциации писателей»;



QuickTime player — медиапроигрыватель QuickTime 7 обеспечивает такие функции, как оцифровка видео изображений, полученных от цифровой видеокамеры с интерфейсом FireWire; быстрая передача по электронной почте и демонстрация видео изображений на пользовательской домашней странице .Mac; оперативное изменение размеров видео и аудио файлов с объемным звуком; дружественный пользовательский интерфейс управления аудио и видео; полупрозрачные пиктограммы управления воспроизведением.

Для управления окнами предусмотрены такие функции как закрытие окна приложения (рис. 2.58, 1), свертывание окна (рис. 2.58, 2), управление размером окна (рис. 2.58, 3), изменение представления содержания (рис. 2.58, 4), масштабирование размера изображения (рис. 2.58, 8).

Панель Dock включает в себя также стеки, что позволяет быстро открывать часто используемые файлы и программы прямо из Dock. В Mac OS X Snow Leopard предусмотрено три готовых стека: Документы, Загрузки и Программы (см. рис. 2.57, 11). В стеке Загрузки сохраняются все загруженные из Internet файлы, чтобы их было проще найти. Файлы, загружаемые в Safari, Mail и iChat, поступают прямо в стек Загрузки. Когда загрузка файла заканчивается, стек помещает новый файл на самый верх, чтобы его было легко найти.

Стеки автоматически разворачиваются веером (рис. 2.58, 9) или отображаются в виде стопки, в зависимости от количества объектов в стеке. Пользователь может указать, какой вид отображения стека следует использовать, а также изменить порядок сортировки объектов в стеке.

Файловая система HFS Plus. В иерархической файловой системе Mac OS X (HFS Plus, или HFS+) каждый файл занимает две части (*fork*), одна из которых может отсутствовать: ветви данных (*data fork*) и ветви ресурсов (*resource fork*). Ветвь данных — это собственно содержание файла, например, текст, изображение, видео или звук. В ветви ресурсов содержится служебная информация о файле — даты создания и редактирования, информа-



Рис. 2.58. Механизмы управления окнами и стеками:

1 — кнопка закрытия приложения (в данном случае — Finder); 2 — кнопка свертывания окна; 3 — кнопка увеличения размера окна; 4 — управление представлением содержания (иконки, список, таблица и т. д.); 5 — левая панель окна приложения (навигации); 6 — окно поиска; 7 — правая панель (представления); 8 — «ползунок» масштабирования; 9 — элементы «всеобщего» представления стека Загрузки; 10 — масштабирование изображений окна

ция о программе-создателе, иконка файла, а если этот файл — программа, то соответствующий код. Различие ресурсов и данных условно — информация файла может быть помещена как в ветвь данных, так и в ветвь ресурсов. Обычно в ветвь ресурсов помещаются те данные, которые ограничены по размеру и количеству значений.

Том ФС HFS Plus делится на секторы (иногда называемые логическими блоками), которые обычно имеют размер 512 байт. Секторы группируются в блоки размещения (кластеры), которые содержат от одного до нескольких секторов и количество которых определяется размером тома (до 2^{32} блоков).

На типичном томе HFS Plus создаются 9 управляющих структур (рис. 2.59). Такие из них, как HFS boot blocks, Volume

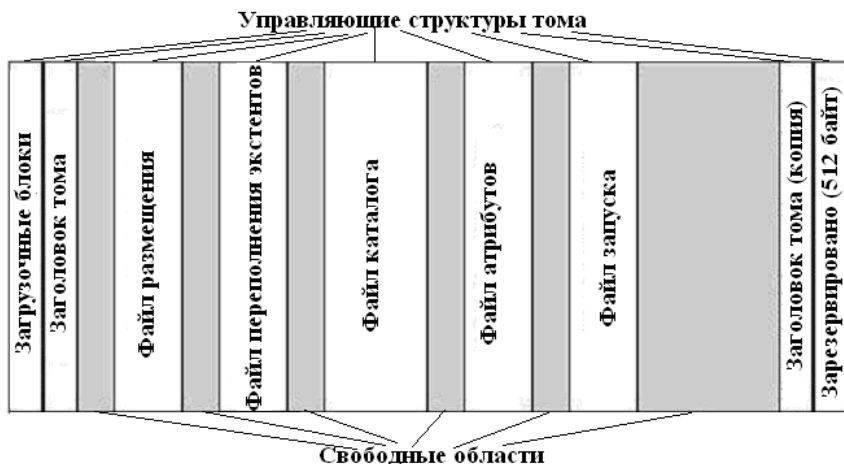


Рис. 2.59. Структура тома с файловой системой HFS Plus

Header, Alternate Volume Header, находятся по фиксированным адресам. Остальные могут помещаться в произвольных секторах тома. Перечислим данные структуры:

- загрузочные блоки (HFS boot blocks) — секторы 0 и 1;
- заголовок тома (Volume Header) — сектор 2. Содержит основные сведения о томе — размер блоков размещения, время создания тома, адреса размещения других структур тома (Catalog File, Extent Overflow File и пр.);
- файл размещения (Allocation File), который содержит информацию о свободных и занятых блоках размещения. Каждый блок представлен одним битом, причем значение «1» соответствует занятому блоку, а «0» — свободному. Иногда такую структуру называют битовая карта (bitmap). Файл может иметь произвольный размер и не обязательно размещается в смежных блоках тома;
- файл каталога (Catalog File) — В-дерево, содержащее записи для всех файлов и папок, размещенных на томе;
- файл переполнения экстенгов (Extents Overflow File) содержит информацию о дополнительных экстентах файлов, не поместившихся в основные записи файлов в каталоге;
- файл атрибутов (Attributes File) содержит 3 типа записей, имеющих длину по 4 Кбайта:

- записи встроенных атрибутов (Inline Data Attribute records), содержащие непосредственно ряд коротких атрибутов;
- записи атрибутов ветвей данных (Fork Data Attribute records), содержащие ссылки на несколько (до восьми) экстенстов, в которые могут быть размещены атрибуты большой длины;
- записи расширения атрибутов (Extension Attribute records), которые содержат атрибуты, не поместившиеся на восьми экстенстах атрибутов ветвей данных.
- файл запуска (Startup File) предназначен для использования другими ОС (не Mac OS), которые не поддерживают ФС HFS Plus;
- копия заголовка тома (Alternate Volume Header) размещается на предпоследнем секторе тома;
- последний сектор тома зарезервирован для использования в дальнейшем.

Контрольные вопросы

1. Что такое дисковая операционная система?
2. Какие операционные системы вам известны?
3. Какие версии DOS вам известны?
4. Какие компоненты входят в состав DOS? Что такое ядро DOS?
5. Что такое командный процессор и какие функции он выполняет?
6. Что такое драйвер?
7. Как происходит начальная загрузка DOS?
8. Что представляет из себя файловая система MS DOS?
9. Какие существуют правила для задания на диске имени файла и каталога? Что такое задание файла по маске?
10. Как строится полное имя файла и каталога?
11. Какие функции выполняют файлы `autoexec.bat` и `config.sys` и какова их типичная структура?
12. Какие основные внутренние команды DOS вы знаете?
13. Какие вам известны основные внешние команды DOS?
14. Как создать каталог и файл в MS DOS?
15. Что такое Windows и какие бывают версии Windows?
16. Какие преимущества имеет Windows?
17. Какие требования предъявляются к компьютеру при установке на нем Windows?

18. Какие функции выполняет Windows? Как запускаются программы в Windows?
19. Как производится запуск Windows и выход из Windows и какие бывают режимы работы у Windows?
20. Как изменить размеры и положение окна в Windows?
21. Что такое Диспетчер Программ Windows и какие функции он выполняет? Меню Диспетчера Программ.
22. Что такое Диспетчер Файлов Windows и какие функции он выполняет? Меню Диспетчера Файлов.
23. Как создать программную группу и программный элемент Windows?
24. Какие основные приложения Windows вы знаете? В какой программной группе они находятся и как их запустить?
25. Что такое Панель Управления Windows? Где она расположена и какие функции выполняет?
26. Что такое `pic`-файл и как производится его создание и редактирование?
27. Что такое пиктограмма Windows? Как она создается? Укажите библиотеки пиктограмм в Windows.
28. Как упорядочить пиктограммы в Windows?
29. Как переключиться от одного приложения к другому, затем завершить задачу в Диспетчере Программ Windows?
30. Что такое буфер обмена Windows?
31. Что такое ОС Windows 95? Какие функции она выполняет?
32. Как запустить оболочку Windows 95 и как из нее выйти?
33. Какие бывают версии Windows 95? Какие преимущества и недостатки имеет ОС Windows 95? Какие требования предъявляются к компьютеру при установке Windows 95?
34. Что такое Рабочий стол и какие его элементы вы знаете?
35. Что такое Панель Задач и Панель управления Windows 95? Каково их назначение? Что такое Проводник и каковы его функции?
36. Что такое Папки и Ярлыки и как они создаются и удаляются?
37. Что такое окна Windows 95 и каковы их основные элементы?
38. В чем состоит документо-ориентированный принцип Windows 95?
39. Каковы особенности файловой системы в Windows 95?
40. Какие важнейшие приложения содержит Windows 95?
41. Какие программы можно использовать для обслуживания диска и восстановления файловой системы в Windows 95, а какие нет?
42. Какие программы позволяют работать в Windows 95 так же, как в Windows 3.1?

Задания

1. Определите версию DOS на своей машине.
2. Проверьте, правильно ли установлены дата и время на вашей машине? При необходимости произведите их коррекцию.
3. Создайте каталог `primer` на диске `c :` и создайте в нем файл `proba.txt`. Затем удалите этот файл и каталог.
4. Вставьте дискету, выведите ее оглавление и затем выполните предыдущее задание.
5. Прочитайте и проанализируйте содержимое файлов `autoexec.bat` и `config.sys`. Скопируйте файл `config.sys` с жесткого диска на дискету.
6. Выйдите из Norton Commander и выведите оглавление диска `c :`, затем перейдите в каталог, где находится Norton Commander, и запустите его.
7. Проверьте свой диск программой `chkdsk` на наличие потерянных кластеров. Затем запустите программу `ScanDisk` и проверьте свой винчестер.
8. Выведите карту ОП вашего компьютера и проанализируйте ее. Затем введите команду очистки экрана.
9. Проведите диагностику своей машины программой `MSD`.
10. Создайте тестовую программную группу `tes_group` и в ней тестовый программный элемент `Windows`. Удалите тестовый программный элемент и программную группу.
11. Измените описание программной группы и программного элемента.
12. Измените вид пиктограммы `Windows`. Создайте свою пиктограмму в редакторе `IconEdit`.
13. Измените положение и размеры окна в `Windows`. Увеличьте окно на весь экран и сверните его обратно в пиктограмму.
14. Запустите приложение `Windows` и вернитесь обратно в `Windows`.
15. Откройте одновременно несколько окон в `Windows` и скопируйте программный элемент из одного окна в другое, затем переместите программный элемент из одного окна в другое.
16. Создайте PIF-файл в редакторе `PIF-Editor` для любой программы.
17. Запустите программу с помощью PIF-файла.
18. Создайте тестовый каталог на диске `c :` в диспетчере файлов `Windows`. Скопируйте в этот каталог группу файлов. Удалите эти пробные файлы, а затем и пробный каталог.
19. Свяжите в Диспетчере Файлов текстовый файл с соответствующим текстовым редактором, графический файл — с графическим редактором.
20. Измените цвета и оформление в `Windows`, установите дату и время в панели управления. Установите хранитель экрана и время задержки его появления в `Windows`.

21. Проверьте переключение клавиатуры РУС/ЛАТ. Проверьте, правильно ли установлены стандарты и драйвер принтера в панели управления Windows.
22. Запустите справочную систему Windows, найдите гипертекстовые ссылки и воспользуйтесь ими. Запустите учебник Windows. Выйдите из учебника и из справочной системы Windows.
23. Запустите Windows 95 и осуществите выход из оболочки Windows 95.
24. Создайте ярлыки основных программ (Word, Excel, Works) и положите их на Рабочий стол. Проверьте, запускаются ли эти программы с помощью ярлыков. Измените внешний вид и название ярлыка.
25. Создайте Папку, измените ее название. Включите в Папку документы.
26. Поместите основные программы (Word, Excel, Works) в меню кнопки Пуск. Запустите эти программы. Произведите поиск файла и папки на диске с : с помощью меню кнопки Пуск.
27. Используйте Корзину для удаления пробных Ярлыка и Папки. Восстановите из Корзины удаленные объекты. Очистите Корзину.
28. Запустите Панель управления и измените параметры оформления Рабочего стола.
29. Запустите основные программы на исполнение с помощью инструмента Мой компьютер. Скопируйте файлы и папки на дискету.
30. Запустите основные программы на исполнение с помощью Проводника.
31. Откройте окно, измените его размеры и место положения. Сверните и разверните окно, закройте окно. Перетащите пробный документ из окна на Рабочий стол. Измените вид содержимого окна.
32. Введите прикладные программы Disk Commander и Windows Commander в пункт Меню Приложение кнопки Пуск в Панели Задач.
33. Определите параметры файла в инструменте Мой компьютер и с помощью Проводника.
34. Произведите поиск заданного файла или папки на диске (дискете) с помощью меню кнопки Пуск.

Глава 3

ОПЕРАЦИОННЫЕ СИСТЕМЫ КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ — МНОГОПОЛЬЗОВАТЕЛЬСКИЕ МНОГОЗАДАЧНЫЕ

Многопользовательские многозадачные ОС в связи с необходимостью обеспечения мультипрограммирования и обеспечения многопользовательского режима обработки данных впервые были разработаны для *больших ЭВМ (mainframes)*.

Первая функционально полноценная ОС — OS/360 — была предложена фирмой IBM в качестве оболочки ЭВМ IBM/360. Разработка и внедрение ОС позволили разграничить функции операторов, администраторов, программистов, пользователей, а также существенно (в десятки и сотни раз) повысить производительность ЭВМ и степень загрузки технических средств. Версии OS/360/370/375 — MFT (мультипрограммирование с фиксированным количеством задач), MVT (с переменным количеством задач), SVS (система с виртуальной памятью), SVM (система виртуальных машин) — последовательно сменяли друг друга и во многом определили современные представления о роли ОС в общей иерархии *систем управления данными и задачами при обработке данных на ЭВМ*.

Ранние версии OS/360 были ориентированы на пакетную (batch processing) обработку информации — входной поток заданий (на МЛ, МД или перфокартах) подготавливался заранее и поступал на обработку в непрерывном режиме. В дальнейшем возникли расширения OS/360/375, допускающие диалоговую обработку данных с терминалов пользователя, последняя из версий (OS SVM) фактически предоставляла в распоряжение пользователя «виртуальную персональную ЭВМ» с полной мощностью вычислительной установки IBM/360/375. ОС других семейств

(поколений), например RSX (для PDP/11 DEC) или Unix, с самого начала ориентировались на интерактивное взаимодействие с пользователями.

Относительно Unix следует отметить, что она в настоящее время является самой популярной ОС (если судить *не по количеству, а по качеству ЭВМ*, на которых она установлена, а также учесть, что MS DOS функционально является подмножеством Unix). Unix-подобные системы известны для:

- больших машин (Digital/PDP, Digital/VAX, Hewlett-Packard 4000/9000, Honeywell 6070, IBM/370, Amdahl 470);
- персональных ЭВМ и рабочих станций (SUN SparcStation, IBM PowerStation, DECStation, систем на процессорах Zilog/Z8000, систем на процессорах Motorola/68000/68040, в том числе Apple Macintosh, систем на процессорах Intel/80286/386/486/Pentium).

Наиболее распространенные версии Unix:

- Bell Labs AT&T — версии System 6, 7, III, V;
- FreeBSD (версии 4.2, 4.3) — реализация Unix для персональных компьютеров, выполненная исследовательской группой вычислительных систем Калифорнийского университета в Беркли. Версии, выпускаемые этой группой, обозначаются BSD (от Berkeley Software Distribution);
- фирменные версии — SCO (Santa Cruz Operation), ISC (Interactive System Corporation), A/UX, SunOs, AIX, Xenix, Ultrics, HP-UX и пр.

Широко распространены также системы, разработанные под влиянием концепций Unix, — MS DOS, Windows 95/NT, OS/2.

В настоящей главе будут кратко рассмотрены операционные системы OS/360, RSX, Unix, Linux.

3.1. Операционные системы OS/360/370/375

Вычислительные машины ряда ЭВМ IBM/360 (ЕС ЭВМ)

Вычислительные машины ряда ЭВМ IBM/360 (в СССР была разработана аналогичная серия — ЕС ЭВМ) представляли собой семейство программно-совместимых машин третьего поколения. Архитектурно машины включают:

- главный процессор (CPU — Central Processing Unit);
- оперативную память (Main storage, Core storage);

- каналы устройств, обеспечивающие операции обмена данными между памятью и внешними устройствами независимо от процессора (channel processor);
- набор внешних устройств ввода-вывода (devices), выполняющих обмен информацией между внешними носителями и каналами.

Для ряда ЭВМ IBM/360 (ЕС ЭВМ) характерно наличие каналов — специализированных процессоров, позволяющих освободить процессор от выполнения операций ввода-вывода и тем самым повысить скорость обмена с внешними устройствами. В машинах ряда ЭВМ IBM/360 (ЕС ЭВМ) с помощью каналов обеспечивается параллельная работа процессора и внешних устройств, а также параллельное выполнение операций ввода-вывода с несколькими внешними устройствами.

Система программного обеспечения ЕС ЭВМ включала в себя операционные системы, комплексы программ технического обслуживания, пакеты прикладных программ. Эта система открытая, т. е. ее состав может пополняться, обеспечивая развитие технических средств, совершенствование методов обработки информации, расширение сфер применения.

Основные сведения о функционировании ОС

Главными функциями ОС являются: управление задачами и управление данными. Эти функции реализуются через формализованное средство описания данных и заданий — язык управления заданиями (JCL).

На рис. 3.1 схематически изображен порядок прохождения заданий в ОС ЕС. Первоначально задания, представляющие собой некоторые тексты (см. ниже), считываются и обрабатываются программой системного ввода, осуществляющей интерпретацию операторов JCL. При отсутствии ошибок очередное задание помещается во входную очередь, ожидая освобождения требуемой области оперативной памяти ЭВМ и других ресурсов. При наличии требуемых ресурсов программа инициатор-терминатор выделяет раздел памяти, размещает первую из программ задания, присоединяет необходимые входные-выходные наборы данных и передает ей управление. После завершения работы программы при отсутствии других пунктов задания осуществляется завершение задания, уничтожение временных наборов данных,

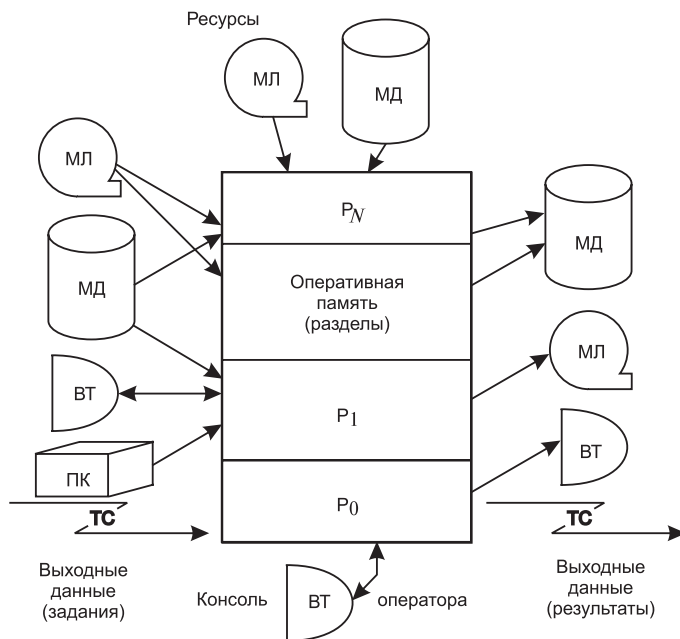


Рис. 3.1. Общая структура прохождения задач в ЭВМ:

МД — накопители на магнитных дисках; МЛ — накопители на магнитных лентах; ТС — телекоммуникационный канал; ВТ — видеотерминалы; ПК — перфокарты; P_i — разделы (сегменты оперативной памяти) ($i = 0, 1, \dots, N$)

заккрытие выходных наборов, вывод информации на внешние носители, освобождение ресурсов и устройств ЭВМ.

Результирующие сообщения задания помещаются в выходную очередь на системную печать.

Язык управления заданиями

Язык управления заданиями (JCL) ОС ЕС включает следующие основные операторы, приведенные в табл. 3.1 (см. также табл. 1.4).

Задание представляет собой совокупность операторов, помещенных между заголовком задания (JOB-карты) и пустым оператором (или следующей JOB-картой), и должно содержать хотя бы один оператор шага задания (EXEC-карту), соответствующий некоторой выполняемой в данном шаге программе. Каждый на-

Таблица 3.1. Основные операторы языка управления заданиями (JCL-операторы, карты)

Название оператора	Формат оператора
Заголовок задания	//ИМЯ JOB [ОПЕРАНДЫ] [КОММЕНТАРИИ]
Заголовок шага задания	// [ИМЯ] EXEC [ОПЕРАНДЫ]
Определения (описания) данных	// [ИМЯ] DD [ОПЕРАНДЫ]
Пустой оператор	//
Комментарий	//* [ТЕКСТ]

Примечание. В квадратные скобки включены необязательные компоненты.

бор данных (НД), вводимый или выводимый программой шага задания, задается картой описания данного (DD-оператором).

Для описания конструкций JCL-операторов обычно используется *нотация IBM*, которая включает следующие конструкции:

- *< > угловые скобки* (или двойные кавычки """), обозначающие элементы программы, определяемые пользователем (<идентификатор>, <список параметров>, <условие> и пр. В соответствующих местах реальной программы будет находиться идентификатор переменной и т. д.);
- *[] квадратные скобки*, ограничивающие синтаксическую конструкцию, обозначают ее возможное отсутствие;
- *{ } фигурные скобки*, содержащие вертикально расположенный список, означают обязательный элемент, одно из значений которого должно быть выбрано из этого списка;
- *... горизонтальное многоточие*, следующее после некоторой синтаксической конструкции, обозначает последовательность конструкций той же самой формы, что и предшествующая многоточию конструкция. Например, = { <выражение> [, <выражение>] ... } обозначает, что одно или более выражений, разделенных запятыми, может появиться между фигурными скобками;
- *вертикальное многоточие* означает пропуск некоторой части программы, представленной в примере.

Каждый оператор DD имеет имя (ddname), используемое для ссылок на этот оператор в программе и других целей. Иногда группа DD-операторов может иметь одно имя, находящееся в карте первого из операторов, а в остальных операторах группы поле имени не заполнено. Такие наборы данных называются

сцепленными, рассматриваются ОС как один набор данных и должны иметь одинаковую организацию, формат и длину записей (см. ниже).

Имеется ряд стандартных и/или зарезервированных в системе dd-имен:

JOBLIB — библиотека задания; оператор размещается непосредственно после карты JOB и описывает одну или несколько (сцепленных) библиотек исполнительных программных модулей, один или несколько из которых вызываются в задании;

STEPLIB — библиотека шага задания. Оператор аналогичен JOBLIB, но действие его распространяется только на данный шаг;

SYSUDUMP — НД, в который при аварийном завершении выводится дамп памяти — содержимое основной памяти, в которой выполняется задание, а также ряд дополнительных данных о состоянии системы;

SYSIN — системный вход; содержит исходную программу транслятора, исходные данные пользовательской программы, управляющие карты утилит; весьма часто размещается во входном потоке (см. ниже);

SYSPRINT — системный вывод; содержит сообщения трансляторов, диагностические сообщения утилит и служебных программ; как правило, помещается в очередь, направленную на выходной принтер;

SYSUT1, SYSUT2, ..., SYSUTN — рабочие НД трансляторов, утилит, редакторов, входные/выходные наборы данных утилит.

Основные параметры операторов управления данными. DD-операторы, как и другие JCL-операторы, могут содержать *ключевые* и *позиционные* параметры (подпараметры). Ключевые имеют вид: ИМЯ = ЗНАЧЕНИЕ, где ИМЯ — зарезервированное наименование параметра, а ЗНАЧЕНИЕ — выбираемое из зарезервированных (или произвольно), задаваемое программистом значение параметра.

Порядок ключевых параметров в операторах и их разрешенное отсутствие не играют роли. Позиционные параметры имен не имеют, и их смысл определяется позицией в конструкции JCL. Они, как правило, разделяются запятыми и отсутствие параметра требует наличия соответствующего разделителя. Наиболее часто употребляются параметры DSNAMES (DSN), DISP, UNIT, VOLUME (VOL), SPACE, LABEL, DCB. Ниже приводятся краткие описания форматов и основных подпараметров данных конструкций.

Параметр *DSNAME* — имя НД и его вид (постоянный или временный).

Параметр задает имя набора, по которому осуществляется его поиск на устройствах ЭВМ или с которым он создается. Для временных наборов данных имя может не указываться либо должно начинаться с символов &&.

Параметр *DISP* — текущее и будущее состояние набора данных, имеет следующую структуру. В то время как сам параметр *DISP* является ключевым подпараметры, указанные в скобках, — позиционные:

$$DISP = \left(\begin{Bmatrix} NEW \\ OLD \\ SHR \end{Bmatrix}, \left[\begin{Bmatrix} KEEP \\ DELETE \\ PASS \end{Bmatrix} \right], \left[\begin{Bmatrix} KEEP \\ DELETE \\ PASS \end{Bmatrix} \right] \right).$$

Первый подпараметр — состояние, может иметь значения:

NEW — набор создается в данном пункте задания. При значении *NEW* необходимо указание также *UNIT* и, как правило, параметров *DCB* и *SPACE* (см. ниже);

OLD — НД существует и программа шага задания получает монопольный доступ к нему;

SHR — то же, но доступ происходит совместно с другими, активными в данный момент, заданиями.

Второй подпараметр (диспозиция):

KEEP — сохранить НД после завершения задания;

DELETE — удалить набор после завершения шага задания;

PASS — сохранить в течение задания, но удалить после завершения последнего шага задания.

Третий подпараметр (условная диспозиция) может иметь значения *KEEP*, *DELETE*, *PASS*, но определяет, что должно происходить с файлом при аварийном завершении. Если параметр отсутствует, временные НД уничтожаются.

Параметр *UNIT* — указывается при создании новых НД или при использовании существующих сохраненных наборов. Имеет вид:

Групповое имя: *UNIT* = (тип устройства, [число устройств]) адрес.

Оба подпараметра — позиционные.

Групповое имя: *TAPE* — магнитные ленты; *SYSDA* — магнитные диски.

Тип устройства: 5010 или 2311 — любой из НМД с пакетами емкостью 7,25 Мбайт; 5061 или 2314 — НМД с пакетами емкостью 29 Мбайт; 5066 или 3330 — НМД емкостью 110 Мбайт; 5010 — накопитель на МЛ и т. п.

Адрес устройства — трехзначное шестнадцатеричное число, состоящее из номера канала ввода-вывода и номера устройства (табл. 3.2).

Таблица 3.2. Примеры типичных адресов устройств

Адрес	Устройство
00A, 00C	Устройства ввода с ПК
00E, 00F	Алфавитно-цифровые построчные печатающие устройства (АЦПУ)
190, 191...	Накопители на МД емкостью 7,25 Мбайт
130, 131...	НМД 29 Мбайт
151, 152...	НМД 100 Мбайт
280, 281...	Накопители на МЛ
0C0, 0C1...	Видеотерминалы ЕС 7920
4C0, 4C1...	Видеотерминалы ЕС 7970
01F	Пишущая машинка (консоль оператора)

Подпараметр «число устройств» — десятичное целое число.

Параметр *VOLUME* указывает имя тома на МЛ или МД, на котором должен быть размещен или найден набор данных. Наиболее употребительной является структура параметра, указывающая на серийный номер, записанный в физической метке тома:

VOL = *SER* = имя_тома.

Параметр *SPACE* используется при размещении нового НД на диске и характеризует выделяемое для этого пространство памяти. Чаще всего задается в виде

$$SPACE = \left(\begin{array}{c} \text{TRK} \\ \text{CYL} \\ \times \text{ЁЊЁЇ} \end{array} \right), \quad (\text{КОЛИЧЕСТВО } [, \text{ПРИРАЩЕНИЕ}]),$$

где первый, обязательный, параметр указывает, в каких единицах измеряется выделенная память:

- TRK — в дорожках;
- CYL — в цилиндрах;

— ЧИСЛО — размер блока (в которых измеряется объем выделенной памяти).

В зависимости от типа устройства единицы измерения имеют различные количественные характеристики:

- для ЕС 5061 — в дорожке 7294 байт, в цилиндре 20 дорожек, в пакете 200 цилиндров;
- для ЕС 5066 — 400 цилиндров по 19 дорожек, каждая длиной 13 Кбайт.

Размер блока не должен превышать длину дорожки. КОЛИЧЕСТВО и ПРИРАЩЕНИЕ определяют объем первичного экстента (непрерывной области) в указанных единицах и вторичного экстента, выделяемого при переполнении первичного. Всего может быть выделено 16 вторичных экстентов, после чего происходит аварийное завершение задания.

Параметр DCB содержит сведения, необходимые для обработки записей, из которых состоит НД; как правило, должен указываться для новых наборов данных. Имеет структуру:

DCB = (ключевые подпараметры).

Всего имеется 23 подпараметра, некоторые взаимоисключающие; ниже перечисляются наиболее употребимые:

$$DSORG = \left\{ \begin{array}{l} PO \\ PS \\ DA \end{array} \right\} \text{ — организация НД,}$$

где PO — библиотечный НД или состоящий из именованных модулей, разделов; PS — последовательный набор данных простейшей организации; DA — набор прямого доступа.

$$RECFM = \left\{ \begin{array}{l} F \\ FB \\ V \\ VB \end{array} \right\} \text{ — формат записи,}$$

где F — записи фиксированной длины, не объединенные в блоки;

V — несблокированные записи переменной длины;

FB — сблочные записи фиксированной длины;

VB — сблочные записи переменной длины;

LRECL = ЧИСЛО — логическая длина записи, длина записи в байтах;

BLKSIZE = ЧИСЛО — длина блока записей в байтах (для форматов FB, VB). Блок записей, или физическая запись, есть массив данных, передаваемый за одно обращение из буфера оперативной памяти на внешний носитель, или наоборот. Длина блока, как правило, не превышает длины дорожки (для МД) и не ограничена для МЛ;

BUFNO = ЧИСЛО — количество буферов в оперативной памяти, отводимых под считывание блоков НД. Длина буфера равна длине блока. С увеличением BUFNO уменьшается количество обращений к внешним носителям при работе программ, однако увеличивается расход оперативной памяти.

Параметр LABEL=(ЧИСЛО, ТИП) — используется для указания порядкового номера файла, расположенного на МЛ, и типа меток МЛ (например, NL — лента без меток; SL — стандартные метки и т. п.).

Перечисленные параметры используются в общих случаях — при размещении наборов данных на конкретных устройствах и типах устройств. В частных случаях, когда НД размещены во входном или выходном потоке ОС (см. рис. 3.6), применяются следующие параметры:

* — позиционный параметр, помещенный сразу после DD в DD-операторе и означающий, что во входном потоке далее находятся данные, входящие в данный набор;

SYSOUT=КЛАСС — означает, что набор данных является выходным и помещается в выходную очередь указанного КЛАССА (как правило, на соответствующее печатающее устройство).

Наконец, параметр DUMMY в DD-предложении символизирует «пустой НД» и является своеобразной «заглушкой» для входных/выходных НД.

Утилиты ОС IBM/360

Основные операции по обработке файлов ОС ЕС осуществляются служебными программами-утилитами. Наиболее употребительными являются следующие: IEBCGENER, IENRTPCH, IENMOVE, IENDASDR, IENINITT, хотя в состав ОС ЕС входит более 50 таких программ.

Информационные потоки. На рис. 3.2 приведена типовая структура информационных потоков в утилитах и общепринятые обозначения входных/выходных НД (см. также рис. 1.25).

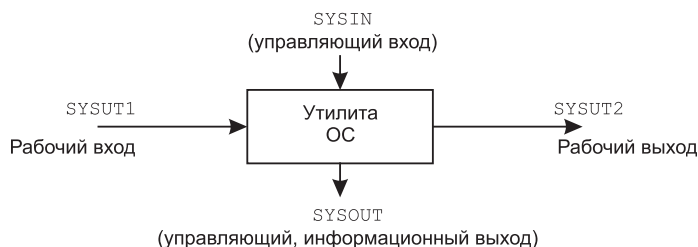


Рис. 3.2. Схема информационных потоков в утилитах

Управляющий вход представляет собой файл, обычно размещаемый во входном потоке и состоящий из одной или нескольких *управляющих карт*, которые содержат списки параметров утилиты, как *ключевых*, так и *позиционных*. Управляющая карта задает режим работы, адреса обрабатываемых файлов, количество преобразуемых записей файла, тип преобразования, формат представления выходного результата и т. д.

Информационный выход содержит сообщения об ошибках в управляющих картах и при обработке данных, протокол работы (число считанных и выведенных записей, содержимое обрабатываемых файлов, имена обрабатываемых файлов и т. д.).

Рабочий вход (может быть несколько) — это обычно входной файл утилиты с исходными данными.

Рабочий выход (также может быть несколько) — выходной файл (файлы) с результатами работы.

Назначение и функции утилит

IEBGENER — создание НД на входных устройствах, копирование последовательных, прямых НД и разделов библиотечных наборов.

На рис. 3.3 приведен пример задания на копирование массива перфокарт, установленного на устройстве с адресом 00С на МД с именем MASTER, в качестве раздела NATEXEC ранее существовавшего библиотечного НД ADA4.SOURCE;

IEHPTPCH — выдача на печать или перфорацию последовательных или библиотечных НД (рис. 3.4);

IEHPRGM — универсальная утилита по созданию, уничтожению НД, каталогов, оглавлений и других системных таблиц;

IEHMOVE — копирование библиотечных, последовательных, прямых НД с МД на МЛ и с МЛ на МД, а также вывод на перфокарты и ввод с перфокарт.


```

ОС ЕС ЭБМ.
//GEN JOB
//G EXEC PGM=IEBGGENER
//SYSUT1 DD UNIT=00C
//SYSUT2 DD
UNIT=SYSDA,DISP=OLD,DSN=ADA4.SOURCE(NATEXEC),
// VOL=SER=MASTER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//

```

Рис. 3.3. Пример JCL-операторов:
IEBGGENER — копирование перфокарт на МД

```

//PTP JOB MSGLEVEL=(1,2)
//PT EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD SYSOUT=A
//SYSUT1 DD *
ПРИМЕР ПЕЧАТИ ПК *)
//SYSIN DD *
PUNCH STOPAFT=2
ОС ЕС IEBPTPCH 63-09.81 PUNCH STOPAFT=2
IEB4231 USE STANDARD FORMAT
IEB4251 EOF ON SYSIN **)
IEB4281 REQUESTED RECORDS WRITTEN FOR SDS
IEB4441 0000000002 INPUT DATA RECORDS READ
IEB4451 0000000002 OUTPUT DATA RECORDS WRITTEN
IEB4461 0000000000 TITLE RECORDS WRITTEN
IEB4421 END OF JOB IEBPTPCH
IEB4431 HIGHEST CONDITION CODE WAS 00
ПРИМЕР ПЕЧАТИ ПК *** )

```

Рис. 3.4. Пример распечатки, иллюстрирующей работу IEBPTPCH:

*) — входной рабочий набор SYSUT1, содержащий три перфокарты с текстовой информацией, из которых, согласно содержимому управляющего входа SYSIN, должно быть считано две и выведено на печать (согласно JCL-карты SYSUT2);

) — протокол работы, также выводимый на печать; *) — выходной рабочий набор — копия двух перфокарт, выведенная также на печать

Выводимые утилитой данные оформляются в стандартном формате, одинаковом для МД, МЛ и перфокарт, что удобно для архивации данных.

На рис. 3.5 приведены карты задания на копирование библиотечного НД IZMIRAN с МД MASTER на МЛ с именем HELP01, имеющую стандартные метки, в качестве файла номер 5;

```
//CDT JOB
//C EXEC PGM=IEHMOVE
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,10)
//TAPE DD
UNIT=5010,DISP=SHR,LABEL=(,SL),VOL=SER=HELP01
//DISK DD UNIT=5061,DISP=SHR,VOL=SER=MASTER
//SYSIN DD *
COPY
PDS=IZMIRAN,FROM=5061=MASTER,TO=5010=(HELP01,5)
//
```

Рис. 3.5. JCL-операторы утилиты IEHMOVE

INHEDASDR — утилита обслуживания пакетов магнитных дисков, форматирование (инициализация) тома, присвоение имени, выявление дефектных дорожек и размещение альтернативных; создание дампа (образа) диска на МЛ — сохранение МД; восстановление МД по образцу на МЛ; прямое полное копирование одного диска на другой;

IEHINITT — инициализация МЛ со стандартными метками — запись в физическое начало ленты файла-метки, содержащего имя тома и необходимую информацию, а также признака конца МЛ, или закрытия ленты.

В ОС ЕС представлены средства *каталогизированных процедур*, предназначенных прежде всего для облегчения использования утилит системы. В специализированных системных библиотеках хранятся заранее подготовленные и соответственно оформленные «заготовки заданий» на выполнение той или иной утилиты или другой программы (совокупности программ), в которых предусмотрены *формальные именуемые параметры*, значения которых задаются путем подстановки *фактических параметров*, указанных пользователем при запуске процедуры. Например, задание на IEHMOVE (см. выше) могло быть результатом следующего обращения к некоторой процедуре CDT (копирование с МД на МЛ) с формальными параметрами:

- T — имя выходной МЛ;
- V — имя входного МД;
- D — имя копируемого набора данных;
- L — тип меток МЛ;
- F — номер файла на ленте.

START CDT V=MASTER,T=HELP01,D=IZMIRAN,L=SL,F=5.

Имена формальных параметров произвольны, определяются системным администратором и должны быть известны пользователям ОС.

3.2. Операционные системы RSX (ОС РВ)

Операционная система RSX (ОС РВ — операционная система с разделением времени) была разработана для машин среднего класса PDP-11 и VAX (фирма Digital Equipment Corporation — DEC). Основная особенность управляющих вычислительных машин типа PDP-11 заключается в том, что взаимодействие между всеми устройствами, входящими в состав комплексов, включая процессор, оперативную и внешнюю память, осуществляется с помощью единого унифицированного интерфейса, получившего название «Общая шина» (ОШ).

В дальнейшем данная архитектура была заимствована для построения персональных компьютеров. Общая шина является каналом, через который передаются адреса, данные, управляющие сигналы на все устройства комплекса. Физически ОШ представляет собой высокочастотную магистраль передачи данных, состоящую из 56 линий.

Процессор использует установленный набор сигналов для связи с памятью и с внешними устройствами, благодаря чему в системе отсутствуют специальные команды ввода-вывода.

Все устройства комплекса подключаются в ОШ по единому принципу. Некоторым регистрам процессора, регистрам внешних устройств, которые являются источниками или приемниками при передаче информации, на ОШ отводятся адреса. В программах адреса регистров устройств рассматриваются как адреса ячеек памяти, что позволяет обращаться к ним с помощью адресных инструкций. Так, программирование операций вывода данных на внешнее устройство практически сводится к пересылке этих данных по определенному адресу.

VAX-11 — более развитая машина, чем PDP-11. Это 32-битовая машина с адресным пространством свыше 4 Гбайт. Она по архитектуре похожа на PDP-11, но имеет два шинных адаптера — адаптер общей шины и адаптер массовой шины. Все совместимые с общей шиной периферийные устройства могут быть подключены к ней, тогда как высокоскоростные устройства могут

быть подключены к массовой шине через собственные контроллеры. VAX — сокращение от англ. слов Virtual Address eXtension — «виртуальное адресное расширение», т. е. машина имеет виртуальную память и многозадачность.

В отечественной практике аналогами данных машин являлись системы СМ ЭВМ — Семейство Малых ЭВМ.

Следует отметить, что в случае ЕС ЭВМ большинство используемых ОС представляли собой семейство систем с расширяющимися от поколения к поколению возможностями и в основном совместимых между собой по формату данных, программ, утилит, компиляторов, имеющих стандартный язык управления заданиями.

Для СМ ЭВМ наряду с широко использовавшейся операционной системой разделения времени (ОСРВ) были распространены ОС ДИАМС, TSX, Unix и другие несовместимые и разнородные по указанным аспектам.

Некоторые основные понятия, связанные с функционированием ОСРВ (RSX)

Пользователь — лицо, осуществляющее запуск, контроль, остановку некоторого вычислительного процесса, протекающего независимо от других, использующего как монопольно выделяемые, так и общие ресурсы. Обязательно выделяемым ресурсом пользователя является терминал ЭВМ, используемый им для выполнения перечисленных функций. Пользователь должен быть зарегистрирован в системе с указанием имени, фамилии, пароля, идентификатора. Пользователи разделяются на *привилегированных и обычных*.

Первые имеют доступ ко всем ресурсам, типам программ, команд, операций, вторые — нет. Тип пользователя задается при его регистрации администратором системы.

Пользовательский идентификатор (UIC — User Identification Code) — код, состоящий из двух чисел G и N, относящий пользователя к некоторой группе и присваивающий ему номер в группе. При организации данных на внешних носителях (МД, МЛ) файлы группируются по UIC и каждая образованная группа получает статус каталога (словаря, оглавления, директории) файлов, находящегося в распоряжении данного пользователя.

Задача — соответственно оформленный и зарегистрированный в системе исполнительный модуль.

Физические устройства ЭВМ регистрируются в ОС, соответствующей фактической конфигурации ЭВМ и идентифицируются кодами: *групповой идентификатор, порядковый номер в группе, символ «двоеточие»* (табл. 3.3).

Таблица 3.3. Идентификаторы основных устройств

Идентификатор	Устройство
DKx :	Накопитель на МД с емкостью 5000 блоков по 512 байт (2,5 Мбайт); x — порядковый номер
DMx :	Дисковый накопитель емкостью 20 000 блоков (10 Мб)
DPx :	МД емкостью 40 000 блоков (20 Мб)
MTx :	Накопитель на МЛ емкостью 20 тыс. блоков
TTx :	Терминал ЭВМ
LPx :	Построчное печатающее устройство
Файл	Набор данных на внешнем носителе

Типичная физическая организация последовательного файла на МД представляет собой размещение логических записей переменной длины, разделяющихся стандартными или специально оговоренными символами-терминаторами, в физических блоках стандартной длины (0,5 Кбайт). В общем случае одна запись может размещаться в нескольких блоках.

Обозначение файла — совокупность символов, идентифицирующих файл и используемых ОС для определения адреса на внешних носителях, состоящая из следующих компонент:

устройство : [g, n] имя.расширение; версия,

где устройство — идентификатор устройства;

[g, n] — каталог (UIC);

имя — выбираемое пользователем наименование НД (не более 8 символов);

расширение — идентификатор файла (не более 3 символов), используемый для группирования файлов в типы.

Некоторые стандартные типы файлов, используемые в ОС и пользовательскими программами:

.ftn — текст программы на Фортране;

.bas — текст программы на ЯП Бейсик;

- .cmd — командный файл;
- .tsk — исполнительный модуль;
- .txt — текстовый файл.

Пример: DP0:[1,7]ADABAS.TSK;1 — это программный модуль с именем ADABAS, размещенный в директории [1,7] на МД с адресом DP0:.

Сокращенное наименование файла может состоять только из имени. При этом устройство и [g,n] берутся из системных умолчаний или пользовательских назначений; расширение задается стандартным типом файла; версия — максимальная (последняя) из существующих.

Спецификация файлов — соглашения о кратком групповом обозначении некоторой совокупности обрабатываемых, переименовываемых, удаляемых, копируемых и пр. файлов.

В спецификации файлов используются символы маскирования «*» и «?», вносимые в компоненты обозначения файла, причем «*» соответствует допустимой строке символов, а «?» — одному допустимому символу.

Примеры:

[*,*]*.TSK;2 — все файлы задач во всех директориях, 2-й версии;

[1,5]ADA*.SYS — файлы директории [1,5] с именем, начинающимся с ADA, расширением SYS, 1-й версии;

[5,5]SYSTEM.?? — файлы с именем SYSTEM, имеющие 2-символьные расширения.

Командные языки — средства общения пользователя с системой, используются для построения команд (или командных строк), а также командных файлов. Основными являются языки MCR (монитор команд терминала) и DCL (командный язык DEC).

Команды MCR имеют следующую общую структуру:

имя [/ключ 1/ключ 2...], где имя — 3-символьный зарезервированный идентификатор. Ключи определяют особенности выполнения команды и могут иметь параметры.

Основные команды (вводятся пользователем с терминала на подсказку MCR> или >):

hel имя/пароль — регистрация входа пользователя в систему с UIC-именем, многопользовательской защитой и закрепление за ним терминала;

mon устройство/ключи — монтирование устройства, закрепление за пользователем устройства;

run имя задачи — запуск программы на исполнение;

abo имя задачи/кл — прекратить работу задачи;
 set — команда изменения системных параметров;
 ufd — создание директории на устройстве;
 dmo устройство — демонтаж устройства, отсоединение от пользователя (терминала);
 bye — регистрация выхода пользователя из системы с многопользовательской защитой.

Утилиты ОС RSX — стандартные программы, предназначенные для обслуживания устройств, МД, МЛ, наборов данных и их групп. Активизация утилиты в любом из командных языков осуществляется путем набора и выполнения строки следующей структуры:

имя [/ключи] [файл-1] [= файл-1]] устр-во-1 [/подключи] устр-во-2 [/подключи]

Здесь имя — 3-символьное наименование утилиты;

ключи (подключи) — управляющие параметры;

файл (устройство)-1 — выходной поток информации (результаты);

файл (устройство)-2 — входной поток информации.

Некоторые наиболее употребимые утилиты:

FMT — утилита форматирования тома на диске — создания структур данных для физического размещения файлов ОС PB;

INI — присвоение метки тома на диске;

BAD — анализ и запоминание адресов физически поврежденных блоков на диске;

PIP — утилита работы с файлами.

Наиболее употребимый формат команды:

>PIP файл-1 = [файл-2] [/подключи] устр-1

Примеры:

>PIP LP1:=DP2:[*,*]/LI — вывод на печать списка всех файлов на DP2.;

>PIP DP1:=DP0:[5,5] — копирование содержимого директории [5,5] диска DP0: в текущую директорию DP1.;

>PIP DP:/FR — показ числа свободных блоков на устройстве DP0;

>PIP DP2:[3,3]*.TSK;*/DE — удаление всех программных модулей из [3,3] устройства DP2.

BRU — утилита быстрого копирования МД на МЛ с целью архивизации и восстановления архива с МЛ на МД.

Пример:

>BRU /NOI/BAC:IVAN1 P1:=MT0: — копирование на устройство DP1: образа диска с именем файла на ленте IVAN1 с ленты на устройстве MT0.;

BCD — анализ и распечатка содержимого ленты, подготовленной с помощью BRU, например:

>BCD LP:=MT: — вывод оглавления МЛ, установленной на MT: на печать.

DMP — вывод содержимого блоков информации некоторого файла на экран или печать, например:

>DMP LP:=HOUSES.D/AS/BL:1:2 — определяет распечатку первых двух блоков файла HOUSES.LOD в коде ASCII.

ACNT — программа работы с учетным файлом пользователей, в котором по каждому зарегистрированному пользователю представлены данные, имя, статус, UIC и уровень доступа к данным. ACNT позволяет вставлять, удалять, изменять эту информацию.

Командный файл — совокупность командных строк, оформленная в виде файла с расширением имени .CMD. Файл обрабатывается *процессором командных файлов*, который распознает и отправляет на исполнение командные строки MCR и DCL, а также анализирует и выполняет операторы и директивы процессора командных файлов.

Некоторые из операторов и директив:

.МЕТКА: — символьный идентификатор, позволяющий осуществлять передачу управления к определенной командной строке;

.ASKN, (.ASKS) — запрос к оператору терминала на ввод числового (символьного) значения, с проверкой на соответствие некоторым пределам;

.SETN (.SETS) — присвоить числовой (символьной) переменной значение;

.GOTO МЕТКА — переход на метку;

.IF — проверить, удовлетворяет ли переменная одному или нескольким условиям;

.BEGIN - .END — блок командных строк, выполняемый полностью при передаче управления на строку .BEGIN;

.EXIT — выход из блока;

.STOP — останов процессора командных файлов.

Запуск командных файлов осуществляется путем ввода пользователем команды: @ИМЯ_ФАЙЛА. Обозначение командного файла должно иметь расширение .COM.

Возможности процессора командных файлов, следовательно, позволяют готовить на данном языке некоторые «программы», осуществляющие вычисления, диалог с пользователем, вызов команд DCL, MCR, утилит и пользовательских программ.

Текстовые редакторы ОС РВ

Необходимо отметить, что средства редактирования могут быть разделены на четыре группы:

- *построчные редакторы*, допускающие обработку текущей строки;
- *экранные редакторы*, манипулирующие с группой строк, размещенных на текущем экране;
- *редакторы набора данных*, позволяющие работать с текущим НД в целом;
- *редакторы группы файлов*, обеспечивающие доступ и обмен информацией внутри совокупности НД одновременно.

В ОС РВ СМ ЭВМ присутствует строчный редактор EDT, однако наиболее распространенным является редактор НД — TED.

Пользователю TED выделяется буфер, в который во время работы помещаются обрабатываемые файлы.

Функциональные возможности TED могут быть разделены на две группы:

- *командный режим работы* (манипулирование файлами и предварительное редактирование) — содержимое буфера визуально недоступно. Команда вводится по подсказке TED>;
- *экранный режим* — точное или окончательное визуальное контролируемое редактирование содержимого буфера с использованием функциональной и управляющей клавиатуры.

Команда TED состоит из кода и параметров (как обязательных, так и нет):

R файл [N1, N2] — прочитать файл от строки N1 до строки N2 в буфер (если параметры опущены, считывается весь файл);

W [P1, P2] [файл] [N] — содержимое буфера от строки P1 до P2 записать в файл после строки N; если опущены P1 и P2, переписывается весь файл; если опущен «файл», запись производится по месту последнего считанного файла; если опущен N, то запись происходит с начала файла;

PR P1 [P2] — распечатать строки буфера в интервале P1—P2 (или строку P1);

D P1 P2 — удалить строки с P1 до P2 включительно;

SU /текст1/текст2/ — контекстная замена цепочек символов в буфере текста;

CL — очистка текстового буфера;

S — перейти в экранный режим;

<Ctrl+Z> — завершение работы TED.

Экранный режим

<Ctrl+K> — вставка пустой строки под отмеченной курсором на экране;

<Ctrl+\\> — вставка пробела в месте, отмеченном курсором, и сдвиг полустроки вправо;

<Ctrl+]> — удаление отмеченного символа и сдвиг правой полустроки влево;

<Ctrl+^> — удаление отмеченной строки;

<RETURN>, <BK> — перевод курсора на строку вниз;

<Ctrl+C> — выход из экранного режима в командный.

3.3. Операционная система Unix

Операционная система Unix — одна из самых популярных в мире операционных систем благодаря тому, что ее сопровождает и распространяет большое число компаний. Первоначально была разработана Кеном Томпсоном — сотрудником фирмы Bell Laboratories концерна AT&T в 1969 г. как многозадачная система для мини-компьютеров и мэйнфреймов. Unix вобрала в себя целый ряд новых идей в области операционных систем. В принципе, она создавалась как операционная система для исследователей. При разработке Unix была поставлена задача создать систему, которая могла бы удовлетворять непрерывно изменяющимся требованиям сотрудников [5].

В 1970 г. Деннис Ритчи вместе с Кеном Томпсоном переписали код системы с машинно-зависимого языка ассемблера (на котором тогда разрабатывались все операционные системы) на язык высокого уровня — Си. Это позволило им написать всего одну версию операционной системы Unix, которую потом можно было компилировать Си-компиляторами на различных машинах. Операционная система Unix стала, по сути дела, мобильной, т. е. спо-

собной работать на различных типах машин практически без перепрограммирования. Кроме того, она позволяет иметь несколько видов Shell, т. е. интерфейсов взаимодействия между ядром и пользователем или интерпретаторов.

В 1974 г. Unix была передана университетам «для образовательных целей», а несколько лет спустя нашла коммерческое применение. С тех пор она выросла в одну из наиболее распространенных операционных систем. Сейчас существуют версии Unix для многих систем, начиная от ПК до суперкомпьютеров.

Для проведения сложных экспериментальных исследований, связанных с большим количеством вычислений над большим объемом данных, требуются значительные системные ресурсы. В этом случае многие Unix-системы позволяют организовать кластер, т. е. многомашинный вычислительный комплекс, где все ресурсы компьютеров (дисковое пространство, память, ресурсы процессора) являются разделяемыми и доступными для любого пользователя в соответствии с его правами. В такой системе существует возможность постоянного наращивания мощности кластера путем подсоединения дополнительных компьютеров, а работа в ней при этом остается для пользователя абсолютно «прозрачной», как если бы он работал на одном компьютере с огромными ресурсами.

Основные компоненты ОС Unix

- core — ядро системы;
- kernel — оболочка ядра системы;
- development system — средства разработки программ;
- manuals — экранные руководства пользователя и программиста;
- VP/ix — эмулятор MS DOS;
- UUCP — средства передачи данных по каналам связи;
- STREAMS — механизм сетевых протоколов;
- TCP/IP — сетевой протокол;
- NFS — сетевая файловая система;
- X Window — средства сетевых графических интерфейсов пользователя (GUI);
- Looking Glass — командная оболочка на базе GUI.

Ядро управляет основными ресурсами (процессор, оперативная память) и периферийными устройствами обмена и хранения данных (магнитные диски, магнитные ленты, принтеры, терминалы, линии связи и т. д.). Одной из функций ядра ОС Unix является программная поддержка файловой системы (ФС). Командный интерпретатор Shell обслуживает терминал пользователя и транслирует команды в запрос к ядру ОС.

Основные понятия, связанные с работой пользователя в ОС Unix

Начало и конец сеанса работ. Каждый пользователь системы имеет:

- *имя пользователя* (для установления взаимодействия пользователей и начисления расходов);
- *пароль пользователя* (для контроля входа в систему и защиты своих данных).

Пользователи могут быть объединены в группы (например, во время работы над совместными проектами) для разделения общих ресурсов, тогда еще есть имя группы пользователей [2].

Один пользователь, называемый *superuser*, является администратором системы (его имя *root*). В частности, он «заводит» (регистрирует) всех прочих пользователей.

Можно сменить свой пароль в любое время:

```
login: peter password: .
> passwd
Changing password for peter
Old password:
New password: ***
Retype new password: ***
> ^D <Ctrl+D> (выход из сеанса)
login:
```

где *>* (подсказка, *prompt*) служит приглашением системы к вводу команды. В конкретной системе форма приглашения может быть изменена, т. е. знак «*>*» может быть заменен на другой символ или строку символов.

Командная строка и формат команд. Командная строка — это последовательность слов, разделенных пробелами. Первое слово командной строки — и есть команда, остальные — параметры.

Типы параметров:

- имя файла = идентификатор (использует символы a—z, A—Z, 0—9, _, ., —);
- опция (ключ) уточняет смысл команды (обычно начинается со знака «минус»).

Например, опция `-al` (может быть со знаком `+` или без него). Смысл опции зависит от команды. Выражение описывает обычно строку символов или является строкой.

Порядок параметров в команде:

```
command options expression filename(s),
```

Некоторые простые команды. На рис. 3.6 приведены результаты выполнения простейших команд — `date` (вывод на экран текущей даты), `who` (получение списка всех активных пользователей) и варианта этой команды `who i am`.

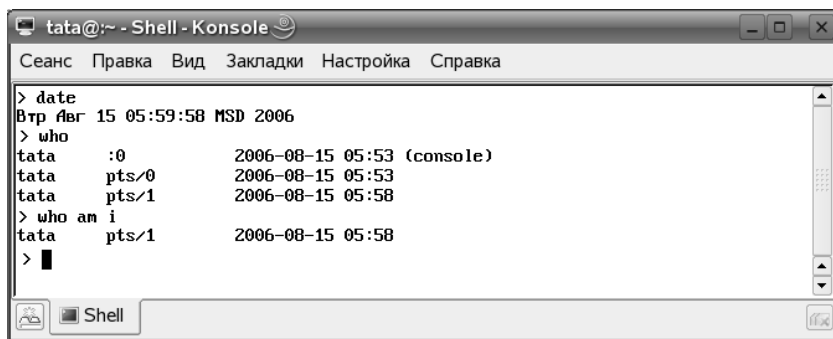


Рис. 3.6. Экран консоли с протоколом выполнения команд `date`, `who`, `who am i`

Другие примеры команд:

```
rm old.news bod.news
rm -fr goodies.c baddies.o
grep -o "mary" people
```

Команды разделяются либо концом строки, либо точкой с запятой, например:

```
> who; date
mary tty0 Feb 14 08:30
sun Feb 14 11:38 1995
>
```

Исправление ошибок при наборе текста команды. При наборе текста команды пользователь может совершить ошибку. Для ее исправления предусмотрены следующие возможности.

Исправление последней буквы:

- `backspace`;
- `^H`;
- `#` (диез).

Исправление последней строки:

`^X`;
`^V`;
`@.`

Приостановка/продолжение вывода на экран

`^S` — приостановка;
`^Q` — продолжение.

Для остановки выполнения команды используются `^C` или клавиша `<Break>` (не работает по линиям связи).

Каталоги и файлы

При регистрации пользователя администратором системы ему назначается собственный каталог пользователя (Home directory).

Правила (соглашения) по наименованию каталогов и файлов.

В именах каталогов и файлов строчные и прописные буквы считаются различными. Символы `'.'` (точка) и `'_'` (знак подчеркивания) не могут использоваться в качестве первой буквы имени. Как правило, имя файла имеет так называемое «расширение», которое характеризует тип файла. Расширение содержит символ, который следует после точки, например:

- `.c` — программа на Си (например, `program.c`);
- `.h` — текст файла-заголовка (header), включаемый в программу на Си;
- `.f` — программа на языке ФОРТРАН;
- `.p` — программа на языке ПАСКАЛЬ;
- `.o` — объектный код, полученный транслятором с любого языка;
- `.a` — библиотечный (архивный) файл.

Использование маскирующих метасимволов *, ? и []. Метасимволы служат для подстановки любых строк и символов в именах файлов и в командах языка заданий Shell:

* — произвольная строка (возможно, пустая);

? — любой одиночный знак;

[C1 C2] — любая литера из диапазона C1—C2 (в стандарте ASCII).

Примеры:

```
1) > ls c?
c1 c2 c3 cs cz
2) > ls c*
c1 c12 c2 c23 c3 cs cs1 cxy cz
3) > ls ?1*
c1 c12
4) > ls *1*
c1 c12 cs1
5) > ls c [12 x y z]
c1 c2 cz
6) > ls c [12 x y z *]
c1 c2 c12 c25 cz cxy
```

Неотображаемые («непечатаемые») символы в именах файлов. Непечатаемыми являются символы со знаком Control (^):

```
^A (<Ctrl+A>) ,;
^[ (<Esc>),
```

и т. п., полученные одновременным нажатием клавиши <Ctrl> и указанной после символа ^ клавиши. Они не видны на экране. В некоторых случаях это может привести к недоразумениям, например команда `ls` может показать файл, а `rm` и другие команды могут не принять имя этого файла (так как часть символов не видна).

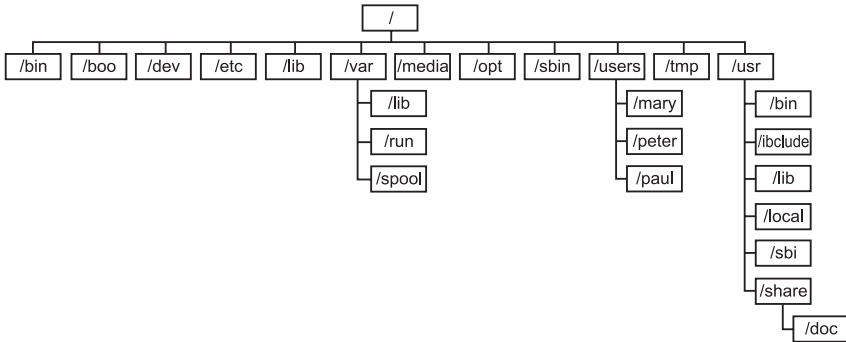
Выход может состоять в использовании метасимволов * в именах (или использование режима `rm -i`):

```
>ls
arron circle square triangle
> rm square
rm: square non-existent (если вместо q в имени ^q)
> Is s* square
> rm s*
>.
```

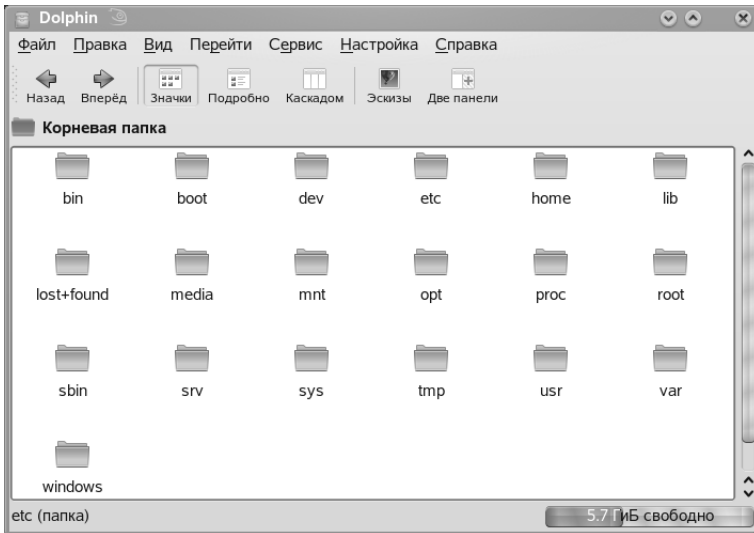
Теперь файл удален.

Структура корневого каталога. Как правило, корневой каталог имеет следующую структуру (рис. 3.7, а), но администратор системы может ее изменить.

На рис. 3.7, б приводится пример распечатки корневого каталога Linux менеджером файлов Dolphin, а в табл. 3.4 — стандартное содержание каталогов Unix-подобных систем.



а



б

Рис. 3.7. Логическая структура ФС Unix (а); отображение содержания корневого каталога (Linux) менеджером файлов Dolphin (б)

Таблица 3.4. Стандартные каталоги файловой системы ОС UNIX

Наименование	Обозначение	Содержание
Корневой каталог	/	Является основой любой ФС UNIX. Все остальные каталоги и файлы располагаются в рамках структуры (дерева), порожденной корневым каталогом, независимо от их физического местонахождения
Команды и утилиты	/bin	Содержит часто употребляемые команды и утилиты системы общего пользования, доступные даже в том случае, если была смонтирована только корневая файловая система. Примерами таких утилит являются: <code>ls</code> , <code>cp</code> , <code>sh</code> и т. п.
Загрузка ОС	/boot	Содержит все необходимое для процесса загрузки операционной системы: программу-загрузчик, образ ядра операционной системы и т. п.
Устройства	/dev	Каталог содержит специальные файлы устройств, с помощью которых осуществляется доступ к периферийным устройствам. Наличие такого каталога не означает, что специальные файлы устройств нельзя создавать в другом месте, просто достаточно удобно иметь один каталог для всех файлов такого типа
Конфигурация	/etc	Содержит системные конфигурационные файлы. В качестве примеров можно привести файлы <code>/etc/fstab</code> , содержащие список монтируемых файловых систем, и <code>/etc/services</code> , который содержит распределение Internet-сервисов по портам
Домашний (необязателен)	/home	Содержит домашние каталоги пользователей. Его наличие в корневом каталоге необязательно, а содержимое зависит от особенностей конкретной UNIX-подобной операционной системы
Библиотеки	/lib	Предназначен для статических и динамических библиотек, необходимых для запуска программ, находящихся в каталогах <code>/bin</code> и <code>/sbin</code>
Съемные носители	/media	Стандартный каталог для временного монтирования файловых систем, например, гибких и компакт-дисков, флэш-накопителей и т. п.
Дополнительное ПО	/opt	Каталог для дополнительного программного обеспечения, устанавливаемого в системе. Обычно в этот каталог устанавливаются программы, не входящие в основной дистрибутив
Суперпользователь (необязателен)	/root	Домашний каталог суперпользователя. Его наличие в корневом каталоге необязательно

Окончание табл. 3.4

Наименование	Обозначение	Содержание
Системное администрирование	/sbin	В каталоге находятся команды и утилиты для системного администратора. Примерами таких команд являются: <code>route</code> , <code>halt</code> , <code>init</code> и т. п. Для аналогичных целей применяются каталоги <code>/usr/sbin</code> и <code>/usr/local/sbin</code>
Редко используемые файлы и команды	/usr	Этот каталог повторяет структуру корневого каталога — содержит каталоги <code>/usr/bin</code> , <code>/usr/lib</code> , <code>/usr/sbin</code> и пр., служащие для аналогичных целей
Подкаталоги /usr	/usr/include	Содержит заголовочные файлы языка C для всевозможных библиотек, расположенных в системе
	/usr/local	Обычно каталог служит для размещения программ, установленных администратором в дополнение к стандартной поставке операционной системы
	/usr/share	Каталог хранит неизменяемые данные для установленных программ. В подкаталог <code>/usr/share/doc</code> обычно добавляется документация ко всем установленным программам
Временные каталоги	/var, /tmp	Используются для хранения временных данных системных (<code>/var</code>) и пользовательских (<code>/tmp</code>) процессов. Каталог <code>/var</code> обычно содержит часто изменяемые системные файлы, например в каталоге <code>/var/log</code> размещаются системные журналы и драйверы

Печать рабочего каталога. Узнать имя текущего (рабочего) каталога можно следующим образом:

```
> pwd (print working directory)
/users/mary
```

В результате исполнения этой команды на экран выводится полное имя рабочего (текущего) каталога `/users/mary`, где `users` — имя вышестоящего каталога в иерархии каталогов ФС.

Изменить текущий каталог можно с помощью команды `cd` (позднее эта команда будет рассмотрена более подробно).

Печать содержимого каталога. Для печати (вывода на экран) каталога используется команда `ls` (`list`).

```
> ls                — текущий каталог;
> ls -a             — -a (all) — все файлы и подкаталоги;
                   — -l (long) — полная информация;
```

```

> ls /           — печать корневого каталога;
> ls -l /        — печать корневого каталога полная;
> ls -l /bin     — часто используемые команды;
> ls -l /bin/bin — редко используемые команды;
> ls /dev        — драйверы устройств;
> ls /users      — каталоги пользователей.

```

Изменение рабочего каталога. Изменение рабочего каталога производится командой `cd` (change directory):

```

> cd /etc        — переход в каталог etc;
> ls -l          — печать команд администратора;
> cd /usr        — переход в каталог usr;
> ls -l          — печать каталога usr;
> cd             — без параметров — возврат
                  к собственному (home) каталогу.

```

Создание каталога пользователем. Рабочий каталог пользователя, являющийся корнем вашей ветви ФС, создается администратором системы. Создать нижележащие каталоги можно командой `mkdir` (make directory).

Уничтожение пустого каталога. Пустой каталог удаляется посредством команды `rmdir` (remove directory):

```

> rmdir progs
[rmdir: progs not empty] — сообщение, поступающее если
каталог не пуст.

```

Печать содержимого файла. Команда `cat` (от concatenate) позволяет объединить несколько (один или более) файлов и направить результат на стандартный вывод (Standart Output), обычно — на экран дисплея, например:

```

> cat /etc/motd — просмотреть файл (message of today)
The system will down at 17:00

```

Этот файл, создаваемый администратором для текущих объявлений, обычно печатается автоматически при входе в систему. Если вы забыли его содержание, можно напечатать еще раз, как в приведенном примере.

Создание файла. Файл может быть создан командой `cat`.

Если не задано имя входного файла, то будет предполагаться стандартный ввод (клавиатура). Одновременно перенаправим вывод в новый (создаваемый) файл:

```

>cat > newfile

```

Здесь стандартный ввод буферизован, т. е. накапливается несколько (обычно 512) знаков и потом осуществляется запись в файл.

Небуферизованный ввод предпочтительнее (на случай сбоев). Для организации небуферизованного ввода используется ключ `-u` (unbuffered).

Пример:

```
> cat
Mary101
Sally113
Jana121
^d [для конца текста (EOF) ]
>
```

Можно добавлять данные в файл:

```
> cat - u >> people
Charlie122
Bill100
^d
>.
```

Просмотр конца файла. Команда `tail` печатает конец файла. По умолчанию — 10 последних строк:

```
> tail /usr/pub/greek
```

Явно можно задать количество (со знаком '-') или номер строки, от которой печатать до конца (со знаком '+'):

```
> tail -3 /usr/greek — три последние строки;
> tail +6 /usr/greek — последние строки, начиная с 6-й.
```

Определение типа файла. Для определения типа файла применяется команда `file`:

```
> file /bin/ls /usr/bin etc/passwd
usr/include/stdio.h
/bin/ls: pure executable — исполняемый;
/usr/bin :directory — каталог;
/etc/passwd: ascii text — текст в коде ASCII;
/usr/include/stdio.h: C program text — текст Си-программы.
```

Копирование файлов. Копирование файлов осуществляется командой `cp` (copy).

Если текущим каталогом является `progs/c`, то, введя команду:

```
> cp /etc/motd message
```

можно создать в текущем каталоге `progs/c` файл `progs/c/message`.

Его можно было бы скопировать в текущий или другой каталог, не меняя имени,

```
> cp /etc/motd progs/c
```

(при этом будет создан файл `progs/c/motd`).

Если `progs/c` — текущий каталог, того же эффекта достигнет команда

```
> cp /etc/motd
```

(с точкой в качестве параметра).

Если второй аргумент команды `cp` — каталог, то в качестве первого можно указать несколько файлов, например:

```
> cp /etc/motd /usr/include/stdio.h progs/c
```

Режим доступа копии файла совпадает с режимом доступа исходного файла.

Перемещение и переименование файлов. Перенос и переименование файлов осуществляются командой `mv` (*move*). Эта команда перемещает файл или каталог из одного места файловой системы в другое. Побочный эффект — переименование файла.

Пример:

```
> mv message mesg
```

```
> ls
```

```
mesg
```

Отличие от копирования состоит в том, что исходный файл будет уничтожен.

Если файл назначения защищен от записи, то система печатает режим доступа и спрашивает подтверждение переноса, например:

```
> cp message mesg
```

```
> chmod 444 message — только чтение
```

```
> mv mesg message
```

```
message:mode 444? y
```

Второй аргумент команды `mv` может быть каталогом, тогда файл переносится под тем же именем:

```
> mv message progs/c
> cd progs.c
> ls
message
```

Как и в случае команды `cp`, может быть перенесено несколько файлов, если второй аргумент — каталог (с точкой в качестве параметра).

Удаление файлов. Удаление (уничтожение) файлов производится с помощью команды `rm` (remove).

Пример:

```
> cd progs/c
> rm motd
> ls
message (нет motd)
```

Одной командой `rm` можно удалить несколько файлов.

Ключ `-i` позволяет использовать интерактивный вариант исполнения команды, требующий подтверждения на удаление каждого файла:

```
> rm-i file1 file2
file1: n (no - нет)
file2: y (yes - да)
```

Можно форсировать уничтожение ключом `-f`, даже если файл защищен по записи, например:

```
> rm -f stdio.h
```

Примечания. 1. Ключи `-f` и `-i` взаимно исключают друг друга.

2. Этими средствами нельзя уничтожить каталог, пока не уничтожены все файлы, входящие в его состав.

Для удаления нескольких каталогов используется ключ рекурсивного уничтожения `-r` (уничтожить все поддиректории):

```
> rm -r progs
```

Вариант:

```
> rm -r * — уничтожить все поддерево.
```

Владелец файла и защита файла

Каждый файл и каталог имеют владельца — обычно это пользователь, создавший их в первый раз. Владелец может затем назначить тип (вид) защиты файла от трех категорий пользователей:

- владельца (самого себя);
- представителей той же группы пользователей, что и владелец;
- всех остальных пользователей системы.

Каждый файл имеет три вида разрешения на доступ (табл. 3.5):

- чтение *r* (*read*) — можно читать (смотреть) содержимое файла или каталога (читать с ключом *-l* в команде *ls*);
- запись *w* (*write*) — можно менять содержимое файла или каталога (создавать или удалять файлы в каталоге);
- выполнять *x* (*execute*) — использовать файл как команду Unix и искать (*search*) в каталоге.

Таблица 3.5. Некоторые варианты назначения доступа к файлам / каталогам (в скобках — восьмеричные числа, кратко характеризующие атрибуты)

Владелец	Группа	Остальные	Краткая запись
<i>rwX</i> (7)	<i>rwX</i> (7)	<i>rwX</i> (7)	777
<i>rw-</i> (6)	<i>rw-</i> (6)	<i>rw-</i> (6)	666
...	...	...	...
<i>r-</i> (4)	<i>r-</i> (4)	<i>r-</i> (4)	444

Пример:

```
> ls-l /bin
-r-xr-xr-x 1bin 1996 Nov.26 12
```

Эта команда показывает режимы доступа.

Команда *chmod* (установка и изменение режима доступа к файлу). Формат команды *chmod* (*change mode*) для установки режима:

```
chmod <режим> <файлы>.
```

Пример:

```
> chmod 644 f1 f2 f3,
```

где 644 соответствует `rw-r--r--`.

Формат команды `chmod` для изменения режима:

```
chmod <изменения> <файлы>.
```

В изменениях используются обозначения:

```
u — user; r — read;  
g — group; w — write;  
o — other; x — execute;  
a — all;  
= — назначить;  
+ — добавить;  
- — отнять.
```

Работа с текстовыми файлами

Она выполняется достаточно простыми средствами печати файлов, поиска строк, замены букв и строк, сравнения файлов и тому подобное. Соответствующие утилиты ОС весьма эффективны для выполнения таких работ по сравнению с обычным текстовым редактором.

Печать файла. Простейший способ печати — это перенаправление стандартного вывода команды `cat` на терминал, имеющий устройство печати.

Пример:

```
> cat file > /dev/lp
```

Однако не все терминалы имеют собственное устройство печати.

В случае общего устройства печати система обеспечивает его коллективное использование, при котором заявка на печать, заставшая устройство занятым, ставится в очередь до момента освобождения устройства, после чего ее требование удовлетворяется автоматически.

В этом случае используется команда `lpr` или `lp`, например:

```
> lpr filea fileb filec.
```

Имеются также команды просмотра очереди заявок на печать `lpstat` и удаления заявки из очереди (`cancel`).

Команды `lpr` и `lp` не обеспечивают разбиение текста на страницы. Это может быть сделано командой `pr` (`prepare`) подготовки файла для постраничной печати, которая предшествует печати.

Пример:

```
> pr myfile | lpr
```

Размеры страницы по умолчанию равны 66 строкам, длина строки — 72 знака.

Ключами `-w` (`width` — ширина) и `-l` (`length` — длина) можно задать другие размеры.

Примеры:

```
> pr -w 132 -h "Conversion program" conv.c | lpr
> pr -l 25 addr | lpr
```

Ключ `-h` (`header`) вводит заголовок печати. Двойные кавычки требуются при наличии в заголовке пробелов, иначе они могут быть опущены.

Разборка и сборка файла. Многие команды ограничивают размеры файла, который они могут обработать. Если файл слишком велик, он может быть разбит на части командой `split`, а после обработки его можно собрать из этих частей командой `cat`. Каждая часть становится независимым файлом с именами по умолчанию `xaa`, `xab`, `xac`, ..., `xzz`. Размер части (по умолчанию) — 1000 строк.

Пример:

```
> split bigfile
```

Можно изменить размер, устанавливаемый по умолчанию, задавая его явно, например 500 строк:

```
> split -500 bigfile
```

Можно задавать имена частей, например:

```
> split bigfile part
```

В этом случае имена файлов будут partaa, partab, ..., partzz.

После необходимой обработки всех или некоторых частей исходного файла его сборка из частей выполняется, например, следующим образом:

```
> cat part?? > bigfile.new,
```

где ?? — метасимволы.

Типичным примером применения технологии разборки-сборки является печать отдельных страниц файла.

Пример:

```
>pr bigfile > bigpr  
>split -66 bigpr  
>lpr xaf xaj
```

Здесь будут напечатаны 6- и 10-я страницы размером по 66 строк исходного файла.

Сортировка текстовых файлов. Утилита `sort` упорядочивает записи файла в алфавитно-цифровом порядке.

Пример:

```
> sort people  
Bill Williams 100  
Henry Morgan 112  
Mary Clark 101  
>
```

Записи отсортированы по первой букве имени, однако можно выполнить сортировку по фамилиям:

```
> sort +1 people  
Mary Clark 102  
Henry Morgan 112  
Bill Williams 100
```

Ключ +1 означает, что одно слово (поле) с начала строки текста (записи) было игнорировано при сортировке.

Сортировка по третьему полю с игнорированием лидирующих пробелов выполняется с использованием ключа `-b` (blank):

```
> sort -b +2 people
Bill Williams100
Mary Clark101
HenryMorgan112
```

Для сохранения результата сортировки в файле используется ключ `-o` (output):

```
sort -o sort people +1 people
```

Для слияния уже отсортированных файлов используется ключ `-m`.

Пример:

```
>sort +1 admpeople > sortadm
>sort +1 hardpeople > sorthard
>sort +1 sortpeople > sortsoft
>sort -m +1 sortadm, sorthard, sortsoft > sortall
>.
```

Имеется возможность удаления дублированных записей, используя ключ `-u` (uniq), а также сортировки по нескольким несмежным полям.

Подсчет строк, слов и знаков в файле. Подсчет числа строк, слов и знаков в заданном файле выполняется командой `wc`.

Пример:

```
> wc people
3951people
>.
```

Ключи `-l` (lines), `-w` (words) и `-c` (characters) могут указать явно объекты счета, например:

```
> wc -l people
3 people
> wc -lc people
1051people
>
```

Поиск строк в файле по образцу (утилита `grep`). Утилита `grep` (аббревиатура от global regular expression printer) осуществляет

поиск по одному или нескольким файлам и печатает все строки, содержащие предъявленный образец текста, на стандартном выводе. В простейшем случае образец задается постоянной строкой знаков. В общем же случае он задается регулярным выражением.

Пример:

```
> grep Henry admpeople hardpeople softpeople
Softpeople: Henry Morgan 112
>
```

или

```
> grep Henry *people
Softpeople: Henry Morgan 112
> -
```

Ключ `-v` (`invert`) предписывает печать всех строк, кроме найденных, например:

```
> grep -v "Henry Morgan" Softpeople
Bill Williams 100
Mary Clark 101
>.
```

Двойные кавычки требуются для размещения в образце пробелов.

Регулярные выражения позволяют вести поиск типа: найти все слова из четырех букв, начинающиеся на `d`, или все слова, кончающиеся на `able`, и т. п.

Рассмотрим примеры задания образцов посредством регулярных выражений.

Знаки `^` и `$` помечают начало и конец строки соответственно:

`"^Genesis"` — найти все строки, начинающиеся словом `Genesis`;

`"eschatus$"` — найти все строки, кончающиеся словом `eschatus`;

`"^Out in cold$"` — найти все строки, равные образцу.

Точка помечает любую букву:

`"^d..."` — найти все слова из четырех букв, начинающиеся с буквы `d`;

`"d...$>"` — то же в конце строки;

"d\". — найти все слова из четырех букв, начинающиеся с d и оканчивающиеся точкой. Знак \ (обратная косая черта) отменяет специальное значение следующего символа.

Квадратные скобки задают списки возможных значений знака:

"^[abcxyz]" — найти все строки, начинающиеся с букв a, b, c, x, y или z;

"^[Dd][a — z][a — z][a — z]" — найти все слова из четырех букв, начинающиеся с D или d, в которых последние три буквы строчные (от a до z).

Фигурные скобки задают количество повторений (замыкание) предыдущего знака:

"^[Dd][a — z]{3}" — то же самое, что и предыдущий пример;

"^[a — z]{3, 5}" — найти все слова, содержащие от трех до пяти строчных букв. Частные случаи замыкания обозначаются специальным образом:

* — для 0 и более раз;

+ — для 1 и более раз;

? — для {0, 1} (ноль или 1 раз).

Пример:

```
> grep ".*" people
```

Эта команда просто напечатает все строки файла.

Примеры использования регулярных выражений.

Уничтожение всех пустых строк в файле:

```
> grep -v "^ $" file > newfile
```

Уничтожение всех строк, состоящих только из пробелов:

```
> grep -v "^ *$" file > newfile
```

Трансляция символов (утилита tr). Утилита tr работает со стандартным вводом и имеет два аргумента, задающих упорядоченные множества знаков, причем каждый знак первого множества заменяется соответствующим знаком второго.

Пример:

```
> tr a-z A-Z <people
MARY CLARK      101
HENRY MORGAN    112
BILL WILLIAMS   100
```

Ключ **-d** позволяет задать множество символов, которые будут уничтожены, например:

```
> tr-d 0-9 < people
Mary Clark
Henry Morgan
Bill Williams
>.
```

Команды сравнения файлов. В процессе разработки программного обеспечения возникает необходимость сравнения версий файла на разных этапах его разработки. Узнать, чем версии отличаются друг от друга, можно с помощью команды `diff`, которая показывает разницу (difference) двух файлов. Сравнение файлов осуществляется по строкам (записям). В результате выполнения команды печатаются строки измененные (c), уничтоженные (d) и добавленные (a) во втором файле-аргументе (по сравнению с первым).

Пример:

```
> cat people
Mary Clark      101
Sally Smith     113
Jane Buily      121
> cat people.new
Mary Clark      101
Sally White     113
James Walker    112
> diff people people.new
2 c 2
<Sally Smith    113
>Sally While    113
3 d 2
<Jane Baily     121
3 a 3
>James Walker   112
```

Знаки `<` и `>` соответствуют лишним или отсутствующим строкам. Эта команда показывает также номера строк, в которых найдены отличия.

Если строки различаются только числом разделяющих слова пробелов, такие различия можно подавить ключом `-b` (blank), например:

```
> diff-b oldfile newfile
>.
```

Другая возможность быстрого сравнения файлов — команда `cmp` (`compare`), реализованная на основе побайтового (побуквенного) сравнения двух файлов.

Пример:

```
> cmp people people.new
people, people.new differ: char 17, line 2
```

В качестве результата печатается число различающихся байтов (букв) и строк (линий).

Ключ `-l` (`long`) позволяет распечатать разницу файлов в виде байтов (адрес и различающиеся значения).

Пример:

```
> cmp -l people people.new
26   123   127
27   155   150
30   150   155
197   60    61
198   60    61
```

Если файлы сильно отличаются друг от друга, их сравнение может быть произведено с помощью команды `comm` (`common`), которая показывает, что в двух файлах одинаковое (общее).

Пример:

```
> cat people
Mary Clark          101
Sally Smith         113
Jane Bailly         121
> cat people.new
Mary Clark          101
Sally White         113
James Walker        112
> comm people people.new
Mary Clark
Sally Smith
Sally White
Jane Bailly
James Walker
```

Результат команды `comm` печатается в три колонки: строки первого файла, отсутствующие во втором; строки второго файла, отсутствующие в первом, и строки, общие для двух файлов.

Можно подавить печать одного или двух столбцов, указывая его номер в виде ключа, например (печать только третьего столбца):

```
> comm -1 2 people people.new
Mary Clark
>
```

Обработка текстовых файлов командой *awk*. *Awk* — это утилита, подобная *grep*. Однако, кроме поиска по образцу, она позволяет проверять отношения между полями строк (записей) и выполнять некоторые действия над строками (генерировать отчеты). Название не является акронимом, оно образовано первыми буквами фамилий авторов (A. V. Aho, P. Y. Weinberger и B. W. Kernighan).

Задание поиска-действия следует синтаксису:

```
/<образец>/ {<действие>}
```

Как образец, так и действие могут отсутствовать. Найденные по образцу строки при отсутствии заданного действия передаются в канал стандартного вывода (на экран). Образец задается регулярным выражением, как и в утилите *grep*. Если образец отсутствует, то обрабатываются все строки.

Рассмотрим примеры действий, которые можно выполнить командой *awk*.

Перестановка полей строки выполняется с помощью ссылки на поле $\$n$, где n — номер поля, например:

```
> cat people
Mary Clark      101
Henry Morgan    112
Bill Williams   100
> awk '{print $2 ", " $1 "^I" $3}' people
Clark, Mary     101
Morgan, Henry   112
Williams, Bill  100
```

Здесь ^I (<Ctrl+I>) — знак табуляции.

Параметры поиска и обработки с помощью утилиты *awk* могут быть заданы в файле, например:

```
> cat swap
{print $2 ", " $1 "^I" $3}
> awk -f swap people
```


Утилита `awk` имеет встроенные образцы и переменные. Образцы `BEGIN` и `END` означают начало и конец файла соответственно. Переменная `NR` (Number of Records) означает число записей (строк) в файле, `NF` — число полей (слов) в записи. Можно использовать переменные, объявленные пользователем.

Пример подсчета среднего значения третьего поля файла `tennis` (программа действий для `awk` находится в файле `average`):

```
> cat > average
{total = total + $3}
END {print "Average value is", total/NR}
^D
> awk -f average tennis
Average value is 8.9
```

Образец поиска в `awk` может содержать условные выражения.

Пример выборки из файла `tennis` всех записей, значение третьего поля в которых не меньше 10:

```
> awk '$3 >=10 {print $0}' tennis
Steve Daniel    11
Hank Parker     18
Jack Austen     14
>.
```

Знак `$0` (доллар-ноль) — это ссылка на всю запись (строку).

В общем случае выражение для условия подчиняется синтаксису, близкому к синтаксису выражений в языке Си. Кроме того, в команде `awk` допустимо указывать отрезок образцов.

Пример выборки всех записей, сделанных с 1996 до 1998 г.:

```
> sort -n -o chard.s chard
> awk '/1996/, /1998/ {if($2 < 8.00 print $0}'
chard.s
1996 7.50 Chateau
1997 7.75 Chateau
1998 5.99 Charles
```

Как видно из примера, в программах действий для `awk` можно использовать управляющие структуры.

Пример цикла для печати полей всех записей файла в обратном порядке:

```
> awk {for (i = NF; i > 0;--i) print $i} fl,
```

где NF — число полей в записи.

Связь пользователь-пользователь

Система Unix предполагает возможность коллективной работы и кооперации пользователей. Это требует развитых средств связи пользователей между собой. В системе имеются следующие возможности коммуникации:

- команда `write` (писать) для немедленной отправки сообщения другому пользователю;
- команда `mail` (почта), реализующая электронную почту.

Посылка сообщений командой write. Командой `write` посылается сообщение указанному пользователю в момент исполнения этой команды. Это означает, что адресат должен быть в этот момент в системе (on line). Немногим нравится, когда сообщение приходит в случайный момент, в особенности во время набора собственного текста. Поэтому вы должны быть уверены в необходимости такого немедленного взаимодействия.

Типичный пример:

```
> write paul
Срочно пришли отзыв на статью!!!
^D
>
```

Сообщение заканчивается вводом знака конца файла <Ctrl+D>.

Адресат увидит у себя на экране:

```
Message from goga tty 00...
Срочно пришли отзыв на статью!!!
EOF
```

Если вы намерены ждать ответа и вести диалог, можно не вводить знак конца файла до конца диалога.

Пример:

```
> write mary
```

Приглашаю сегодня вечером в гости.

Извини, сегодня я иду в театр.

Тогда завтра?

И завтра не могу. Давай на следующей неделе?

Ну, хорошо.

EOF

^D

>

На стороне абонента на экране будет:

```
Message from goga tty 00...
```

Приглашаю сегодня вечером в гости.

Извини, сегодня я иду в театр.

Тогда завтра?

И завтра не могу. Давай на следующей неделе?

Ну, хорошо.

^D

Чтобы узнать, работает ли абонент в системе в данный момент, можно воспользоваться командой `who` (см. также рис. 3.6).

Пример:

```
> who
```

```
goga    tty00    08:30
```

```
paul    tty03    08:31
```

```
boris   tty07    04:12
```

```
mary    tty08    09:01
```

Текст сообщения можно взять из файла, перенаправив стандартный ввод для `write` из этого файла.

Пример (сообщение из файла `message`):

```
write peter < message
```

Текст сообщения можно формировать, выполняя команды внутри текста. В этом случае команде должен предшествовать восклицательный знак, например:

```
> write peter
```

Нужные тебе файлы в каталоге:

```
!pwd
```

```
/users/mary/docs/specs
!  
/users/mary/docs/specs  
^D  
>
```

Ответ команды завершается также восклицательным знаком. Ни команда, ни ответ в текст сообщения не попадают.

Пользователь может запретить получение сообщений на терминал командой:

```
> mesg n  
> ,
```

где *n* — от *no* (нет), и наоборот, разрешить прием командой:

```
> mesg y  
> ,
```

где *y* — от *yes* (да).

Эта же команда без параметров сообщает, в каком состоянии находится терминал пользователя (*no* или *yes*), например:

```
> mesg  
is y,
```

т. е. прием разрешен.

При входе в систему устанавливается состояние *y*.

Электронная почта (*mail*). Электронная почта — это средство, позволяющее пользователям посылать друг другу сообщения, которые накапливаются в почтовых ящиках, реализованных в системе.

При входе пользователя в систему он извещается о наличии почты, например:

```
login: goga password: ...  
you have mail  
>
```

В отличие от команды *write*, в режиме *mail* о приходе почты пользователь извещается только после окончания текущей работы. Поступит сообщение:

```
you have mail (для вас есть почта).
```

Для получения почты необходимо ввести команду `mail` без параметров:

```
> mail
From peter Wed Jim 9 17:58:23 1999
Завтра в 16:30 можно поиграть в преферанс.
Если согласен, сообщи, где встретимся
?
```

Подсказка (?) означает, что система `mail` ждет указания о том, что делать с почтой: печатать, сохранить, уничтожить или выйти из команды `mail`.

Введя знак вопроса, вы получите меню возможных действий (подкоманды):

```
??
q (quit) — закончить;
x (exit without changing mail) — выйти без изменения почты;
p (print) — печатать;
s [file] save — сохранить в файле;
w [file] same without header — то же без заголовка;
- print previous — печатать предыдущее;
d (delete) — уничтожить;
+ next (no delete) — следующее (не уничтожать);
m user (mail to user) — переправить другому пользователю;
! cmd (execute cmd) — выполнить команду;
?
```

Кроме этих десяти подкоманд, можно также ввести `<Ctrl+D>` (то же, что и `q`) или нажать клавишу возврата каретки (`<Enter>`).

При сохранении почты в файле на экране высвечивается следующее письмо, если оно есть:

```
? s from_peter
From peter
.....
?
```

Можно переслать данное письмо другим пользователям, например, используя подкоманду `m`:

```
? m mary boris george
```

Для отправки почты необходимо ввести команду `mail` с параметром (имя пользователя) и текст письма, заканчивая его знаком конца файла (`<Ctrl+D>`), например:

```
> mail peter
Извини, я не смогу играть завтра.
^D
>
```

Отправка ответа может быть произведена при просмотре почты.

Пример:

```
?! mail peter
Извини, я...
^D
```

Для просмотра почты из файла можно ввести команду `mail` с ключом `-f`:

```
> mail -f from_peter
```

Для выхода из режима `mail` необходимо набрать `q`, например:

```
? q
you have mail
>
```

Работа с почтой закончена.

Стандартные файлы

Многие команды работают по умолчанию со стандартными файлами:

- Standard Input (S.I.);
- Standard Output (S.O.);
- Diagnostic Output (D.O.).

Однако есть средства изменения параметров умолчания, т. е. возможность указать другие файлы вместо стандартных. Можно также в качестве D.O. использовать S.O. Эти средства называются перенаправлением (redirection) ввода и вывода.

Перенаправление стандартного вывода. Для перенаправления вывода используется знак `>`.

Примеры:

```
> ls-1 — вывод на экран = standard output;  
> ls-1 > dirconts — вывод в файл dirconts;  
> cat dirconts — вывод на экран.
```

Пробелы вокруг символа «>» необязательны. Возможно перенаправление вывода с добавлением (с дописыванием в файл), обозначается >>.

Пример:

```
> pwd >> dirconts — добавить в файл имя текущего каталога.
```

Перенаправление стандартного ввода. Для перенаправления стандартного ввода используется знак <.

Примеры:

```
> mail — ввод сообщения с экрана;  
> mail < message — ввод сообщения из файла message.
```

Эта возможность используется реже, чем перенаправление вывода. Тривиальный случай перенаправления ввода:

```
> cat < this_file  
> cat this_file  
(это две эквивалентные команды).
```

Можно одновременно перенаправить и ввод, и вывод, например:

```
> cat < left > right
```

Нужно, чтобы left не равнялось right, иначе можно потерять входной файл. Безопаснее использовать знак >>, чем > (т. е. добавление безопаснее, чем запись).

Организация конвейеров команд. Конвейером называется группа команд, объединенных программными каналами. Программный канал образуется назначением стандартного вывода одной команды стандартным вводом следующей команды. Для формирования программного канала используется знак | (вертикальная черта).

Пример 1:

```
> who | wc -l — создание списка активных пользователей
и подсчет их числа (count);
19 — ответ, т. е. 19 пользователей.
>
```

Пример 2:

<code>> ls -l /tmp</code>	<code> grep maryann</code>	<code> sort +3nr</code>	<code> lpr</code>
листинг каталога /tmp	поиск записей, со- держащих строку «maryann»	сортировка (по 4-му полю) най- денных записей	печать упоря- доченного списка

Фильтры. Так называются команды, которые могут получать данные со стандартного ввода и выводить данные на стандартный вывод. Большинство команд является фильтрами, однако есть исключения, например команда `ls` не может работать со стандартным вводом, а команда `lpr` не может работать со стандартным выводом.

Диагностический вывод. Сообщения об ошибках, возникающих при выполнении команд, используют так называемый диагностический вывод. По умолчанию диагностический (как и стандартный) вывод производится на экран, однако он может быть перенаправлен в любой файл. Для этого используется дескриптор файла (целое число), который для стандартных файлов равен:

```
0 — Standard input;
1 — Standard output;
2 — Diagnostic output.
```

Пример:

```
> cat somefile > outfile 2> errfile,
```

где знак `>` эквивалентен `1>`.

Если вы хотите, чтобы сообщения об ошибках нигде не проявлялись, направьте их на `/dev/null`.

Если вы хотите направить сообщения об ошибках туда же, куда производится вывод данных, нужно набрать

```
> cat somefile |& lpr.
```

Обработка команд в фоновом (background) режиме. Обычно команды выполняются в интерактивном (foreground) режиме,

т. е. «пока вы ждете». Однако, если во время выполнения некоторой команды вы хотите выполнять другие команды, то первую команду можно выполнить в фоновом (background) режиме, например:

```
> nroff doc &  
2042  
>.
```

Как видно из примера, для организации фонового выполнения команды `nroff` использован завершающий знак `&`.

Система Unix создает процесс, который выполняется независимо от командного интерпретатора. Ответ 2042 — это идентификатор созданного процесса (PID).

Стандартный вывод фонового процесса лучше перенаправлять в файл (из-за опасности совмещения на экране стандартного вывода интерактивного и фонового процессов), т. е.:

```
> nroff doc > doc.format &  
2042  
>
```

или

```
> nroff doc | lpr &  
2042  
>
```

Чтобы выяснить состояние фонового процесса, следует использовать команду `ps` (process status):

```
> ps
```

PID	TTY	TIME	CMD	
2036	02	0:05	sh	(login — процесс терминала № 2)
2042	02	0:02	nroff doc	(background)
2043	02	0:01	ps	(cat ps)
2050	08	0:03	sh	(login — процесс терминала № 8)

Это показывает, что выполнение команды `nroff` еще не закончилось. Ключи `-l` и `-a` в команде `ps` могут дать больше информации об активных процессах:

- `-l` — информация о родителе, адрес, приоритет и т. д.
- `-a` — информация о всех процессах системы.

Выполнение процессов с низким приоритетом. Рассмотрим организацию выполнения процесса с низким приоритетом (понижением приоритета). Приоритет процесса определяется значением параметра `nice` (приоритет тем выше, чем меньше `nice`).

```
> nice nroff doc > doc.fmt &  
2099  
>.
```

Пусть начальное значение `nice` равно 20. Перед исполнением команды этот параметр автоматически инкрементируется (подвергается приращению). По умолчанию величина инкремента равна 10. Таким образом, приоритет созданного процесса будет соответствовать величине `nice = 30`.

Рассмотрим еще один пример:

```
> nice-5 nroff doc > doc.fmt &.
```

Здесь инкремент задан явно и равен -5. В результате параметр `nice` будет равен 25 и соответственно приоритет будет на 5 единиц выше, чем по умолчанию.

Уничтожение процесса. Для уничтожения процессов служит команда `kill`.

Пример:

```
> kill 2042 (завершить процесс с PID = 2042)
```

или

```
> kill 9 2042,
```

где число 9 — ключ (сигнал) безусловного останова. Другие сигналы из интервала (1, 15), посылаемые системой задаче, можно блокировать, написав соответствующую процедуру реакции на сигналы.

Средства разработки программ

Система Unix обеспечивает богатый набор средств для разработки программ, включающий компиляторы, редактор связей (linker), символьный отладчик, средства ведения программных проектов и разработки языковых процессоров, архивные средства и другие.

Редакторы *ex* и *vi*. Обозначения *ex* и *vi* — два различных имени расширенной версии редактора *ed* (который входит в комплект стандартной поставки системы). Эта программа работает как экранно-ориентированный редактор при обращении по имени *vi* и как строчно-ориентированный редактор при обращении по имени *ex*. Редактор *vi* (аббревиатура словосочетания *Visual Interpretator* — визуальный интерпретатор) может быть включен, а может быть не включен в версию ОС Unix.

Работа с этими редакторами производится в двух режимах: *командном и ввода текста (text entry)*. Переключение в командный режим осуществляется клавишей <Esc>.

*Вызов редактора *vi**

Пример команды:

```
> vi myfile
```

В качестве параметра может быть указано одно или несколько имен файлов (через пробелы) для их последовательного вызова на редактирование. Если имя файла не указано, то появится начало пустого файла (курсор в начале первой строки).

Варианты использования команды:

```
> vi + myfile
```

На экране будет находиться конец файла; курсор — в начале последней строки.

```
> vi +10 myfile
```

Файл будет выведен таким образом, что строка 10 окажется в центре экрана, а курсор будет расположен в начале этой строки.

*Выход из редактора *vi**

Для выхода из редактора используются два метода:

- если вы хотите запомнить изменения — Esc:wq!Enter (нажать на клавишу <Esc>, ввести двоеточие (оно появится в нижней части экрана), затем ввести символы *w*, *q*, восклицательный знак и нажать на клавишу <Enter>);
- если нет — Esc:g!Enter.

Вызов компиляторов. В системе Unix имеются компиляторы с языков Си, Фортран-77, Паскаль и др. Команды вызова компилятора имеют вид *cc*, *fc*, *pc* и т. п. Параметрами этих команд

являются имена файлов с текстами программ на исходных языках. Имена этих файлов должны иметь расширения `.c`, `.f`, `.p` и т. п.

Примеры:

```
> cc program.c
> fc test.f
> pc example.p
```

Результатом работы компилятора является файл исполняемого кода, имеющий по умолчанию имя `a.out`. Если вы хотите дать ему другое имя, это имя следует указать явно, с ключом `-o`, т. е.: `-o <имя_файла>`.

Пример:

```
>fc -o test test.f
>ls test test.f
>
```

Редактор связей. На практике программы создаются из множества отдельно транслируемых модулей, каждый из которых занимает отдельный файл. Результатом компиляции каждого модуля является файл объектного (перемещаемого) кода, имя которого получается заменой исходного расширения `.c` (или `.f`, `.p` и т. д.) на `.o`. Затем все объектные файлы объединяются с помощью редактора связей в единую программу, помещаемую в файл исполняемого кода.

Редактор связей (linker) может вызываться как независимой командой `ld`, так и автоматически при выполнении команд вызова компилятора `cc`, `fc`, `pc` и т. д. В последнем случае эти команды могут иметь несколько параметров-файлов, имена которых могут оканчиваться не только на `.c`, `.f`, `.p`, ..., но и на `.o`.

Файлы исходного текста компилируются, а затем все файлы объектного кода, как полученные в результате компиляции, так и указанные в качестве параметров команды вызова компилятора, передаются редактору связей. Результатом его работы по-прежнему является файл с именем `a.out` (если не указано явно другое имя). При этом, как правило, объектные файлы уничтожаются. Чтобы сохранить их, можно подавить автоматический вызов редактора связей ключом `-c` (только компиляция) в команде вызова компилятора.

Пример:

```
> fc -c test.f check prove.f
> ld /lib/frt0.o *.o -l F77
> ls a.out check.f check.o prove.f prove.o test.f
test.o
>
```

Здесь добавлен файл `/lib/frt0.o` стартового модуля для программы на Фортране (`/lib/crt0.o` для Си) и библиотека подпрограмм `lF77` для Фортрана (`lc` для Си). Могут быть добавлены и другие библиотеки. Обозначение `-lx` является сокращением для `/lib/libx.a` при любом `x`. Следует заметить, что библиотеки указываются последними (не являются ключами команды `ld`).

При автоматическом вызове линкера (редактора связей) стартовый модуль и ряд библиотек вызываются по умолчанию. Чтобы их увидеть, в командах вызова компилятора следует применить ключ `-v`.

Библиотеки программ. Как отмечалось выше, на вход редактора связей могут подаваться не только файлы объектного кода, но и библиотечные файлы, которые оказываются удобным средством хранения объектных модулей, если их становится очень много.

Имя библиотечного файла обычно оканчивается на `.a`. Для создания, пополнения и просмотра библиотечных файлов используется команда `ar` (архив).

Пример создания библиотеки из трех объектных файлов:

```
> ar rcv exam.a test.o check.o prove.o
a - test.o
a - check.o
a - prove.o
>
```

Здесь ключи команды `ar` означают:

- r** - заменить (replace) модули в библиотеке;
- c** - создать (create) библиотеку;
- v** - печатать включаемые модули (verbose).

Теперь мы можем вывести на экран содержимое библиотеки командой `ar` с ключом `t` (table of content):

```
> ar t exam.a test.o check.o prove.o
```

и ссылаться на библиотеку в командах вызова компиляторов или редактора связей, например:

```
> ar t exam.a  
test.o  
check.o  
prove.o  
>
```

Следует помнить, что порядок размещения модулей в библиотеке существенен. Например, если подпрограмма `test` вызывает подпрограмму `check`, то файл `test.o` должен предшествовать файлу `check.o` в библиотеке.

Для выявления и печати таких зависимостей предназначена команда `lorder`.

Менеджеры пакетов. Современные системы на базе Unix/Linux состоят из большого числа зависимых библиотек, исполняемых файлов, скриптов и т. д. Удаление или изменение версии одного из составляющих систему компонентов может повлечь неработоспособность других, связанных с ним компонентов, или даже вывести из строя всю систему.

Большая часть программного обеспечения в Linux изначально поставляется в виде исходных текстов и для установки требуется скомпилировать программу и получить бинарную версию. Для поддержания целостности и обеспечения возможности распространения программ в бинарном виде используются **менеджеры пакетов**.

Пакет содержит собранную программу, информацию о том, какие требуется совершить действия для ее установки, информацию о зависимостях программ, а также ряд других данных в зависимости от вида пакета.

Часто компоненты, используемые различными программами, выделяют в отдельные пакеты и помечают, что для работы ПО, предоставленного пакетом I, необходимо установить пакет II. В таком случае говорят, что пакет I зависит от пакета II (между пакетами I и II существует зависимость).

Отслеживание зависимостей между такими пакетами представляет собой серьезную задачу для любого дистрибутива — некоторые компоненты могут быть взаимозаменяемыми и при удовлетворении тех или иных требований может обнаружиться несколько пакетов, предлагающих затребованный ресурс.

Задача контроля целостности и непротиворечивости установленного в системе ПО еще сложнее. Представим, что некие программы I и II требуют наличия в системе компоненты III версии 2.1. Обновление версии пакета I, требующее обновления компоненты III до новой, использующей новые возможности версии (скажем, до версии 2.3), влечет за собой требование обновления программы II.

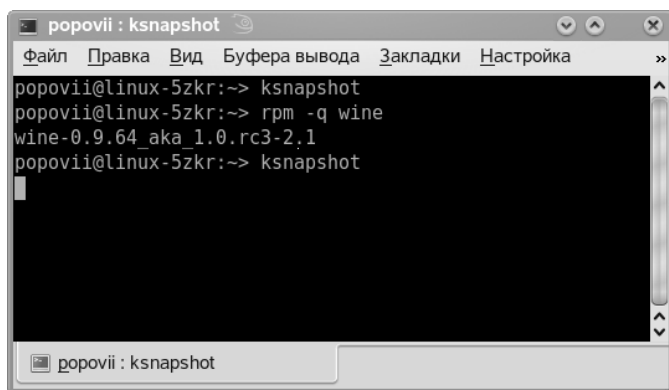


Рис. 3.8. Экран выдачи менеджером пакетов rpm справки о состоянии программы wine

Менеджер пакетов (МП) — это набор инструментов, позволяющих автоматизировать процесс установки, обновления, конфигурирования и удаления пакетов программного обеспечения с компьютера. Обычно менеджер пакетов является основой дистрибутива, обеспечивая полный контроль целостности и работоспособности всей системы.

Менеджер пакетов реализует следующие функции:

- контроль целостности пакетов;
- поддержку установки, обновления и удаления пакетов;
- контроль зависимостей;
- реализацию поиска по доступным/установленным пакетам.

Пакеты собираются в репозитории. Менеджер пакетов обеспечивает пользователю интерфейс для автоматизированного получения пакета, его зависимостей и его установку. Последовательность действий следующая:

- пользователь запрашивает установку пакета;
- МП отслеживает зависимости;

- МП выбирает зависимые пакеты из репозитория;
- МП устанавливает зависимости и требуемый пакет.

Практически каждый дистрибутив Linux имеет свои репозитории, зачастую не совместимые с другими дистрибутивами.

Менеджер пакетов — консольная утилита (рис. 3.8), однако обычно к ней существуют многочисленные графические оболочки. К примеру, МП `dpkg` имеет как консольные оболочки (`apt`, `aptitude`), так и графические `Synaptic`, `Adept`.

Основные менеджеры пакетов:

- `rpm` (Redhat Package Manager) — используется в RedHat Linux, Fedora, SUSE, Mandriva и многих других. Формат пакета: `rpm`;
- `dpkg` — используется в Debian, Ubuntu и многих других дистрибутивах. Формат пакета: `deb`;

`Pacman` используется в ArchLinux, производных и некоторых других дистрибутивах. Формат пакета: `pkg.tar.gz`.

Системное администрирование

В функции администратора системы Unix входит повседневное управление системой во всех аспектах ее существования, таких как подключение новых пользователей, управление файловой системой, изменение конфигурации и др. Следует заметить, что на персональных ЭВМ эти функции могут исполняться прикладным программистом.

Имеется ряд команд, расположенных обычно в каталоге `/etc`, рассчитанных на управление системой, таких как `fsck`, `mount`, `chown` и т. д. Как правило, каталог `/etc` доступен только суперпользователю, так что системный администратор должен обладать правами суперпользователя.

Спецпользователи. Это пользователи, выполняющие действия над системой, недоступные обычным пользователям. Один из них, имеющий неограниченные полномочия, называется суперпользователем и имеет обычно имя `root`. Разные системы могут иметь и других спецпользователей, например пользователя с именем `bin`, обладающих меньшими полномочиями, чем суперпользователь.

Имеются команды, которые может выполнить только суперпользователь, в частности установка даты командой `date`, монтаж файловой системы командой `mount`, создание специальных файлов командой `mknod` и др.

Стать суперпользователем можно несколькими способами. Первый — загрузить систему в режиме единственного пользователя. Другой, применяемый на многопользовательской системе, — выполнить команду `SU` (`superuser`). Администратору системы рекомендуется входить в систему как обычному пользователю и только в случае необходимости становиться временно суперпользователем по команде `SU`.

Пользователи и группы. Имеется два файла с именами `passwd` и `group`, находящихся в каталоге `/etc`, которые содержат информацию о пользователях и группах пользователей, соответственно. Одна запись в файле `passwd` соответствует одному пользователю и имеет следующие текстовые поля, разделенные символом двоеточия:

- имя пользователя;
- пароль пользователя (в закодированном виде);
- целочисленный идентификатор пользователя;
- целочисленный идентификатор группы;
- комментарий, который содержит сведения о месте работы пользователя и может использоваться командой `finger` и учетными программами;
- каталог пользователя;
- интерпретатор команд пользователя.

Пример записи файла `passwd` (с пустым комментарием):

```
mary:KmHulhE:201:10::/users/mary:/bin/csh
```

При наличии комментария его синтаксис определяется учетными программами.

Для некоторых системных программ требуется, чтобы идентификатор суперпользователя был равен нулю, а имя — `root`.

Одна запись в файле `group` соответствует одной группе и представляет собой строку текста со следующими полями, разделенными двоеточиями:

- имя группы;
- пароль группы (в закодированном виде);
- целочисленный идентификатор группы;
- список имен пользователей группы, разделенных запятыми.

Пример записи файла `group` (для группы без пароля):

```
sect2115::10:mary, sas, temp, ges
```

Добавление нового пользователя в системе требует выполнения следующих трех действий, которые обычно реализуются командным файлом с именем `newuser` или `adduser`:

- добавить запись в файл `passwd` с информацией о пользователе;
- создать каталог пользователя, причем пользователь должен быть владельцем этого каталога;
- добавить или скорректировать запись в файле `group` в соответствии с членством пользователя в некоторой группе.

Добавление и коррекция записей в файлах `passwd` и `group` могут выполняться текстовым редактором (если нет командного файла `newuser`).

Каталог пользователя создается суперпользователем (возможно, посредством скрипта `newuser`) и вначале принадлежит ему. Чтобы изменить пользовательскую и групповую принадлежность каталога (и любого файла), используются команды `chown` (`change own`) и `chgrp` (`change group`) соответственно. Их может выполнить только суперпользователь.

Часто новый пользователь забывает свой пароль. Суперпользователь может в этом случае изменить пароль пользователя командой `passwd`, затерев забытый и сообщив пользователю новый:

```
# passwd mary
New passwd: mmm
Retype new passwd: mmm
#
```

Вводимый пароль (здесь — `mmm`) не виден.

Подключение терминалов. Все терминалы, которые могут быть подключены к системе, должны быть описаны в специальном файле с именем `/etc/ttys` (версия 7) или `/etc/inittab`.

Каждому терминалу в этих файлах соответствует одна строка. Форматы файлов `/etc/ttys` и `/etc/inittab` сходны в том, что первый символ является цифрой, нулевое значение которой соответствует отключенному (логически) терминалу, а единичное — подключенному терминалу. Кроме того, формат обоих файлов предусматривает наличие имени спецфайла, соответствующего терминалу (второе поле в файле `/etc/inittab`).

Типичная строка в файле `/etc/inittab` имеет вид:

```
1:t3:c:/etc/getty tty!3 n 0
```

Первое поле имеет подполя, разделенные двоеточием.

При загрузке системы ее последним шагом является запуск начального процесса с номером 1, выполняющего команду `/etc/init`. Команда `init`, обрабатываемая перед выполнением системного стартового командного файла `/etc/re`, просматривает файл `etc/inittab`. Для каждой строки этого файла, начинающейся с ненулевого символа, совпадающего с состоянием начального процесса (при запуске равно единице), команда `init` порождает второй процесс в цепочке (`init-getty-login-shell`).

Порожденный процесс исполняет команду, указанную в четвертом подполе первого поля файла `/etc/inittab` (в рассмотренном примере — `/etc/getty`).

Команда `getty` выдает на терминал, указанный именем спецфайла во втором поле файла `/etc/inittab`, содержимое файла `/etc/issue`, если он существует, и вслед за этим выводит на терминал текст подсказки из записи файла `/etc/gettydefs`, содержащего характеристики терминала (обычно — `login:`). После этого команда `getty` читает имя пользователя и вызывает команду `login`, передавая ей имя пользователя в качестве параметра.

Команда `login` вводит пароль пользователя и после успешной проверки пароля выполняет команду из последнего поля записи данного пользователя в файле `/etc/passwd` (обычно — `/bin/sh` или `bin/csh`), а также устанавливает в качестве текущего начальный каталог пользователя, указанный в предпоследнем поле записи данного пользователя в файле `/etc/passwd`.

Команда `sh` или `csh` командного интерпретатора выполняет стартовые файлы с предопределенными именами (`.profile` для `sh`; `.cshrc` и `.login` для `csh`), выводит подсказку и ждет ввода очередной команды пользователя.

По окончании сеанса (подачей команды `logout` или `^D`), а также в случае неверного пароля управление возвращается в команду `getty`, которая перезапускается повторно, если в третьем подполе первого поля записи файла `/etc/inittab` для данного терминала стоит символ `C` (`continually`); если же в этом поле стоит пробел, то команда `getty` завершается.

Периодическое выполнение заданий. Одной из команд, выполняемых в составе системного стартового командного файла `/etc/rc` начальным процессом, может быть команда `/etc/cron`, создающая постоянный процесс, пробуждающийся периодически.

ски каждую минуту. Этот процесс просматривает записи файла `/usr/lib/crontab` в поисках заданий, которые должны быть выполнены.

Типичный файл `crontab` может выглядеть следующим образом:

```
cat /usr/lib/crontab
0 0 * * * /etc/backup -fsck
0,15,30,45 2-23 * * * /usr/lib/atrun
```

Первые пять полей записи файла `crontab` означают минуты (0—59), часы (0—23), день месяца (1—31), месяц года (1—12) и день недели (0—6, 0 — воскресенье). В каждом из этих полей может быть значение; перечень значений, разделенных запятыми, или границы интервала значений, разделенные минусом. Звездочка означает любое возможное значение.

В приведенном примере первая запись соответствует ежесуточному выполнению (в полночь) сброса и проверки файловой системы в течение всего года, вторая — выполнению программы запуска заданий, запланированных командой `at`, каждые 15 минут с 2 часов ночи до 23 часов вечера ежедневно в течение всего года.

Команда `at` планирует выполнение командного файла интерпретатора `shell` (shell-скрипта), указанного ее последним аргументом в момент времени, заданный ее первыми аргументами, например:

```
at 2300 jun 16 scriptfile_1
```

Указанный файл будет выполнен в 23 часа вечера в указанный день текущего года. Точность времени запуска зависит от периода пробуждения постоянного процесса, выполняющего команду `cron`. Все действия процесса, выполняющего `cron`, фиксируются в учетном файле `/usr/lib/cronlog`, если он существует и открыт на запись в момент выполнения этой команды.

Команду `at` может запустить любой пользователь, чтобы оставить задание на ночное время.

Управление операционной системой. Средства управления операционной системой — аппаратно-зависимы, однако приводимое ниже описание этих средств является типичным для большинства версий ОС Unix.

Операционная система хранится на дисковом томе в некотором заданном формате. Для задания формата используется ко-

манда инициализации тома `/lbin/init` с одним обязательным параметром — именем спецфайла для устройства, на котором находится инициализируемый том. Остальные необязательные параметры могут указать размер логического блока, единицы передачи данных между томом и оперативной памятью и размер загрузочной области в байтах (при отсутствии их значения выбираются по умолчанию).

Каждый том имеет одну загрузочную область, содержащую целое, возможно нулевое, число логических блоков. Загрузочная область полностью находится вне какой-либо файловой системы. Изменение ее размера возможно только при переинициализации тома.

Каждая загрузочная область может содержать только одну ОС (или часть одной ОС). ОС состоит из последовательности кодовых сегментов, расположенных в загрузочной области одного или нескольких томов, причем граница между томами может быть внутри некоторого сегмента.

ОС хранится в загрузочном формате. Помимо загрузочной области, она может располагаться также в ряде обычных файлов, каждый из которых содержит целое число кодовых сегментов, заканчивающихся двумя нулевыми байтами. Этот формат не является загрузочным, однако он может быть преобразован в загрузочный командой `/lbin/oscp`.

Содержимое загрузочной области в действительности состоит из одного или нескольких ОС-файлов. Каждый ОС-файл начинается с заголовка, содержащего флаг загружаемости, номер тома и число томов, занятых операционной системой. Системный загрузчик загружает ОС-файл только в том случае, если флаг установлен в состояние загружаемости. Установку состояния флагов ОС-файлов можно выполнить специальной командой `/lbin/osmark` с параметром, указывающим имя спецфайла для устройства, на которое поставлен том с загрузочной областью. Ключ в этой команде устанавливает флаг в состояние загружаемости или незагружаемости. Упомянутая выше команда `/lbin/oscp` позволяет выполнить следующие действия по копированию сегментов ОС:

- копировать ОС из одной (или более) загрузочных областей тома (томов) в загрузочную область другого тома;
- копировать ОС из обычных файлов в ОС-файлы для создания (опция `-m`, от *merge*) или модификации (опция `-a`, от *add*) ОС в загрузочной области;

- копировать ОС-файлы в обычные файлы для разделения ОС на части (опция — `s`, от `split`) или в один обычный файл (опция `-f`, от `file`).

Для проверки целостности ОС в загрузочной области, а также для контроля добавленных сегментов можно выполнить команду `/lbin/osck`, например:

```
osck -v /dev/rhd
```

Опция `-v` вызывает печать списка имен всех сегментов ОС. Команда проверяет корректность:

- заголовков ОС-файлов;
- списка сегментов;
- контрольной суммы каждого сегмента.

Первоначальная установка ОС или части ОС выполняется командой `optinstall`, а модификация версии ОС — командой `optupdate`. Эти команды следует выполнять в однопользовательском режиме и завершать перезагрузкой системы.

Данные для установки или модификации версии ОС расположены на дистрибутивном томе. Параметром обеих команд является имя (номер) устанавливаемого или модифицируемого программного продукта.

Наконец, имеется возможность загружать несколько разных ОС (или версий одной ОС), используя одну загрузочную область на системном диске (`/dev/rhd`). Этот вариант загрузки выполняет команда `/lbin/chsys` (`change system`), являющаяся командным файлом. Внутри командного файла `chsys` используются команды: `oscp` — для перестройки загрузочной области для новой ОС чтением ее сегментов из обычных файлов и `osck` — для последующей проверки загрузочной области.

Так как команда `chsys` не проверяет, все ли пользователи закончили работу, рекомендуется перед ее выполнением осуществить команду `shutdown`.

Переконфигурирование операционной системы. Параметризуемость операционной системы позволяет оптимальным образом настроить ее для работы на заданных аппаратных средствах и с учетом особенностей использования системы для заданного класса задач. Совокупность значений технических параметров ОС в загрузочной области называют конфигурацией ОС.

Для управления конфигурацией имеется команда `/lbin/uconfig`. Ее необязательный параметр указывает спецфайл уст-

ройства, на котором находится загрузочная область (по умолчанию — /dev/rhd). Будучи поданной без опций, команда `uconfig` показывает текущее значение параметров. Опция `-f <имя_файла>` позволяет установить новые значения нескольких или всех параметров из указанного файла, а опция `-d (default)` устанавливает всем параметрам значения по умолчанию, используя файл /etc/uconfigtab.

Системными параметрами являются (в скобках указаны значения по умолчанию):

- устройство виртуальной памяти (системный диск);
- размер буфера кэш-памяти (1024 байта);
- число буферов кэш-памяти (0, вычисляется динамически);
- длина цепочки буферов чтения (0, вычисляется динамически);
- время активности процесса после интерактивного чтения;
- время резидентности сегмента в памяти перед свопингом на диск (0, вычисляется динамически);
- размер страницы (1024 байта);
- время резидентности страницы в памяти перед свопингом на диск;
- максимальный размер страничного пула виртуальной памяти (0, вычисляется динамически);
- число страниц буфера дисплея, где страница равна 24 строкам дисплея;
- максимальный размер стековой памяти (0, вычисляется динамически);
- минимальная доля страниц в рабочем множестве страниц;
- максимальное число процессов одного пользователя (500).

При выполнении команды `uconfig` для изменения системных параметров нужно быть уверенным, что ОС в загрузочной области совпадает с текущей ОС. В противном случае, результаты выполнения команды будут непредсказуемы.

Загрузка и выключение системы. Загрузка осуществляется, когда на ЭВМ только что включили питание. Обычно процесс загрузки в большей или меньшей степени автоматизирован и заключается в последовательном вызове программ, каждая из которых загружает и запускает следующую. Первая программа, самая простая, загружена всегда и запускается при включении питания ЭВМ автоматически или вручную. В процессе загрузки может потребоваться ответить на вопросы системы, например, касающиеся устройства, на котором находится загружаемая сис-

тема. В завершение процесса загрузки система выполняет командный файл `/etc/rc`, который, вообще говоря, может содержать любые команды, но обычно содержит команды для выполнения следующих действий:

- демонтаж старых файловых систем;
- монтаж новых файловых систем;
- удаление старых журнальных учетных файлов;
- удаление временных файлов;
- запуск процессов `update` и `cron`.

Выключение многопользовательской системы производится выполнением командного файла `/etc/shutdown`, который обычно выполняет следующие действия:

- посылает предупреждающие сообщения всем активным пользователям;
- уничтожает все процессы, кроме процесса консоли;
- очищает все буферы обменов с файлами;
- демонтирует файловые системы;
- выполняет, если нужно, процедуры копирования данных;
- выключает питание ЭВМ, если это позволяет аппаратура, в противном случае питание выключается вручную.

Файловые системы

Файловая система имеет иерархическую структуру каталогов и файлов, включая корневой каталог. Файловая система располагается на устройстве, которое, как правило, является магнитным диском того или иного типа. Если диск достаточно велик, он может быть разбит на несколько логических дисков, тогда на каждом логическом диске может быть размещена отдельная файловая система [2, 12].

Каждая файловая система, прежде чем стать доступной, должна быть смонтирована. Количество файлов в файловой системе ограничено (65 536 для Unix версии 7).

Физическая структура файловой системы (на примере ФС EXT2FS). Каждая файловая система имеет четыре основные части (рис. 3.9):

- загрузочный блок — это самый первый блок диска (блок 0), зарезервированный для системной загрузочной программы;

- **суперблок** — это первый блок собственно файловой системы (блок 1), он содержит основные данные о файловой системе и ее размещении на диске, в том числе — списки свободных *i*-узлов и блоков. От целостности суперблока непосредственно зависит работоспособность файловой системы. Операционная система создает несколько резервных копий суперблока для возможности его восстановления в случае повреждения;
- **индексные дескрипторы** (информационные узлы, индексные узлы, *information node*, или *i-node*, *i*-узлы) — это последовательность блоков вслед за суперблоком. Это специальные структуры, которые содержат информацию об атрибутах и физическом расположении файла. Каждый *i*-узел содержит ссылки на блоки и существует ровно один *i*-узел для каждого каталога или файла в файловой системе;
- **блоки**, которые содержат либо непосредственно данные каталогов и файлов (блоки данных), либо ссылки на другие блоки (косвенные блоки), занимают оставшееся пространство диска.

Суперблок содержит следующие данные:

- размер дискового пространства, доступного файловой системе (в блоках);
- число блоков, зарегистрированных для *i*-узлов;
- имя файловой системы;
- имя тома;
- время последнего изменения;
- время последнего копирования (Backup);
- ссылку на список свободных блоков;
- ссылку на список свободных *i*-узлов.

Каждый *i*-узел имеет длину 64 байта и содержит указатели на блоки, составляющие файл. Структура и атрибуты *i*-узла:

- *Mode* (2 байта) — тип файла, биты защиты;
- *Nlinks* (2 байта) — количество каталоговых записей, указывающих на этот *i*-узел;
- *Uid* (2) — идентификатор владельца;
- *Gid* (2) — номер группы;
- *Size* (4) — размер файла в байтах;
- *Addr* (39) — адресная часть (адреса 10 первых блоков файла и трех косвенных блоков);
- *Gen* (1) — счетчик использования *i*-узла;
- *Atime* (4) — время последнего доступа к файлу;

- Mtime (4) — время последнего изменения файла;
- Ctime (4) — время последнего изменения *i*-узла.

Структура файловой системы представлена на рис. 3.9.

В данном примере блоки на диске имеют длину в 1 Кбайт, длина адреса — 32 бита, что позволяет адресовать 1 Кбайт $\times \times 2^{32} \approx 4,3$ Тбайт. В одном блоке может быть размещено $1024/4 = 256$ адресов других блоков. Для файлов различных размеров используются следующие методы адресации блоков:

- небольшие файлы — адреса первых 10 блоков файла хранятся непосредственно в *i*-узле (указатели 1—10, рис. 3.9), поэтому при размере блока в 1 Кбайт длина такого файла может составлять до 10 Кбайт;
- файлы среднего размера — используется косвенный блок (одинарный косвенный блок — 1x indirect block или просто indirect block) или дополнительный (промежуточный) блок с адресами 256 блоков файла. На него ссылается 11-й указатель адресов в *i*-узле. Длина файла здесь может составлять до 266 Кбайт (10 Кбайт + 256×1 Кбайт = 266 Кбайт);

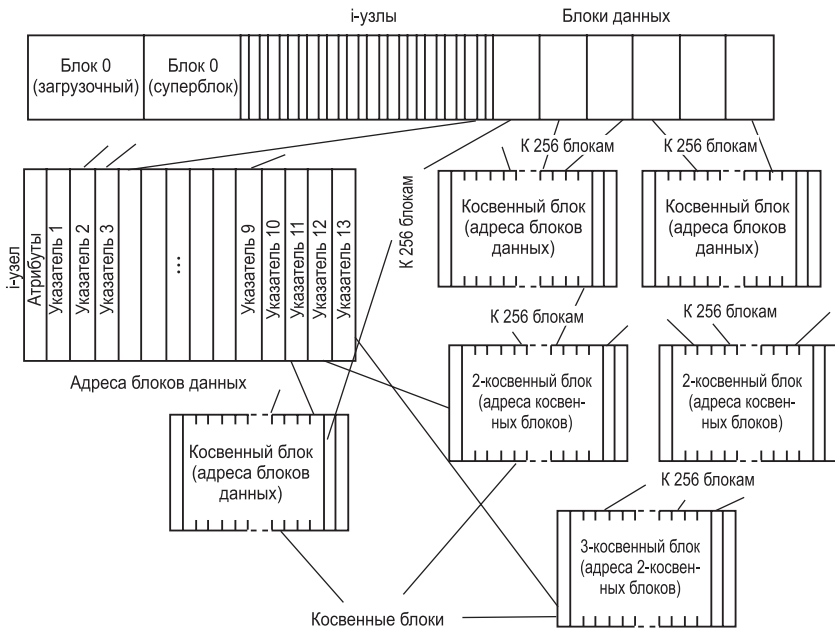


Рис. 3.9. Физическая структура файловой системы ОС Unix V7

- файлы большого размера используют двойной косвенный (дважды косвенный — 2x indirect block или double indirect block) блок с адресами 256 одинарных косвенных блоков. Длина файла может достигать $10 \text{ Кбайт} + 2^8 \text{ Кбайт} + 2^{16} \text{ Кбайт} = 65 \text{ Мбайт}$;
- очень большие файлы адресуются с использованием тройного косвенного (трижды косвенного — 3x indirect block или triple indirect block) блока с адресами 256 двойных косвенных блоков. В этом случае длина адресуемого файла может достигать $10 \text{ Кбайт} + 2^8 \text{ Кбайт} + 2^{16} \text{ Кбайт} + 2^{24} \text{ Кбайт} = 16 \text{ Гбайт}$.

Другие версии системы могут отличаться количеством ссылок в *i*-узле, косвенных блоках и размером блока данных.

Когда система загружается, имеется только одна из файловых систем, называемая корневой. В ней находятся все важнейшие каталоги (*/dev*, */etc*, */bin* и пр.). Все остальные файловые системы должны быть созданы и смонтированы.

Создание и монтирование файловой системы. Команда *mkfs* создает новую файловую систему. Она расположена в каталоге */etc* и имеет два параметра:

```
/etc/mkfs <имя> <размер>
```

Первый параметр является именем специального файла и указывает устройство, на котором создается файловая система. Второй параметр — размер пространства файловой системы в блоках, он используется для определения по некоторым правилам числа блоков после того, как размещены *i*-узлы.

Пример создания файловой системы на флоппи-диске:

```
>/etc/mkfs /dev/flo 2000  
isize = 230
```

Ответное сообщение указывает число блоков, выделенное для размещения *i*-узлов. Далее, чтобы сделать файловую систему известной операционной системе, надо ее смонтировать командой *mount*. Эта команда подключает корневой каталог монтируемой файловой системы в один из каталогов корневой файловой системы. Команда расположена в каталоге */etc* и имеет два параметра:

```
/etc/mount <устройство> <каталог>
```

Первый параметр является именем спецфайла для монтируемого логического устройства, содержащего подключаемую файловую систему. Второй — имя уже существующего каталога, под которым монтируется файловая система.

Пример монтирования вновь созданной файловой системы на гибком диске под каталогом, созданным командой `mkdir` в корне корневой файловой системы:

```
>cd/  
Mkdir floppy 0  
/etc/mount/dev/f10/floppy0
```

Чтобы выяснить, какие файловые системы смонтированы в данный момент, нужно подавать команду `mount` без параметров:

```
>Mount  
/ dev/ f10 on / floppy0.
```

Ответом является сообщение об этих системах (в данном случае — одной).

Оно формируется на основе данных о монтировании файловых систем, хранимых в файле

```
/ etc/ mnttab/.
```

Следует заботиться о том, чтобы права доступа корневого каталога монтируемой файловой системы и каталога, под которым производится монтаж, были одинаковыми во избежание ошибок операционной системы.

Сохранение и восстановление файлов. Независимо от объема данных в системе важно иметь регулярную процедуру сохранения (копирования) файлов, чтобы обеспечить восстановление в случае их аварийной потери. Возможны различные способы сохранения. Наиболее распространенным является еженедельное полное копирование и ежедневное инкрементное копирование (только изменившихся со времени последнего копирования) файлов. При этом файлы копируются (сбрасываются) на специальное внешнее устройство памяти, обычно магнитную ленту, однако это может быть и съемный дисковый том, а для малых систем — гибкий диск. На этом устройстве файлы хранятся в специальном архивном формате.

Восстановление утраченных файлов производится путем их поименного копирования из архивной ленты или тома в файло-

вую систему. Обычно таких файлов немного (например, один или два).

В различных реализациях системы могут быть разные команды сброса файлов в архив и восстановления их из архива. Это может быть пара команд `dump` и `restor`, предназначенных для передачи файлов в архив и обратно. Или это может быть одна команда `cpio` (или `tcio` для кассетной ленты) с опциями `-o` или `-i` для сброса в архив и извлечения из архива соответственно.

Наиболее мощным средством сброса в архив в некоторых реализациях служат команда `backup`, являющаяся командным файлом, использующим команды типа `cpio/tcio` и `fsck`.

Команда `backup` позволяет параметризовать процедуру сброса в архив простым редактированием ее текста, задавая следующие параметры:

- имя каталога сбрасываемой иерархии файлов;
- имя учетного файла процедуры сброса;
- имя даты последнего сброса;
- имя файла с напоминанием смены архивной ленты (если архив не уместится на одной ленте);
- спецификацию архивного устройства;
- имя учетного файла процедуры проверки файловой системы.

Наконец, для сброса на ленту или гибкий диск и обратного восстановления применяется команда `tar` (*tape archive*). В отличие от некоторых перечисленных выше команд, она доступна не только администратору системы, но и любому пользователю. Например, для того чтобы сбросить все файлы текущего каталога на гибкий диск, создавая архив впервые (опция `c` *create*), нужно выполнить команду:

```
tar cf /dev/f10*
```

Опция `f` (*file*) указывает, что следующий параметр является именем спецфайла, соответствующего архивному устройству; `*` — метасимвол, показывающий, что в архив копируются все файлы.

Для просмотра содержимого архива следует употребить опцию `t` (*listing*):

```
tar tf /dev/f10
```

Для извлечения из архива указанных файлов нужно выполнить эту же команду с опцией `x` (*extract*). Например, для вос-

становления всех файлов, имена которых оканчиваются на `people`, нужно выполнить команду:

```
tar xf /dev/f10 *people
```

Файлы с этими именами уже должны существовать в текущем каталоге.

В случае, если в текущем каталоге указанных файлов нет, можно восстановить все файлы из архива в указанном каталоге, например:

```
tar xf /dev/f10 mary,
```

где `mary` — каталог.

Проверка и восстановление структуры файловой системы.

Структура файловой системы, описанная выше в терминах *i*-узлов, блоков, косвенных блоков и суперблока, может быть нарушена и требовать восстановления. Например, при разрушении информации в трижды косвенном блоке могут появиться следующие проблемы:

- некоторый блок может быть вне системы, т. е. не являться частью файла и не быть в списке свободных блоков;
- могут появиться дубли *i*-узлов, т. е. записи, описывающие один и тот же файл дважды;
- некоторый блок может одновременно быть частью файла и быть в списке свободных блоков;
- некоторый файл может существовать, не будучи включенным ни в один каталог.

Однако структура файловой системы обладает некоторой избыточностью, позволяющей восстанавливать отдельные поломки. Вот некоторые виды избыточности:

- блок данных, являющийся каталогом, содержит имена файлов и номера *i*-узлов, т. е. существует *i*-узел, соответствующий этому каталогу, и этот *i*-узел должен быть каталогом, а не обычным файлом;
- блок, включенный в список свободных блоков, теоретически не может быть частью какого-либо файла. Для проверки этого достаточно сравнить список блоков, занятых файлами, и список свободных блоков;
- блок, принадлежащий файлу, должен принадлежать только одному файлу.

Эти и другие виды избыточности использует программа проверки файловой системы, запускаемая командой `fsck` (File System Check). В различных реализациях существуют разные команды проверки целостности файловой системы: `ichck`, `dcheck`, `ncheck` и т. д. Однако их возможности в большей или меньшей степени перекрываются с возможностями команды `fsck`.

Пример выполнения команды `fsck`:

```
/etc/fsck  
** Phase 1 - Check Blocks and Sizes  
** Phase 2 - Check Pathnames  
** Phase 3 - Check Connectivity  
** Phase 4 - Check Reference Count  
** Phase 5 - Check Free List
```

Из примера видно, что выполнение команды `fsck` производится в несколько этапов.

На этих этапах производится следующая работа:

- проверка целостности *i*-узлов (счетчик связи, тип и формат *i*-узла);
- проверка каталогов, указывающих на *i*-узлы, содержащие ошибки; проверка каталогов, на которые нет ссылок;
- проверка счетчиков связей в каталогах и файлах;
- проверка неверных блоков и дублированных блоков в списках: свободных блоков, неиспользуемых блоков, которые должны быть, но не включены в список свободных блоков и, наконец, счетчика общего числа свободных блоков.

Команда `fsck` по умолчанию всегда проверяет корневую файловую систему. Все другие файловые системы проверяются, если их имена занесены в файл `/etc/checklist`.

Следующий пример показывает действия команды `fsck` и администратора в случае обнаружения дубля *i*-узла для файла `/user/src/sys/ux`. Администратор принимает решение удалить этот плохой файл, отвечая в диалоге согласием `y` (`yes`) на вопросы команды.

```
/etc/fsck  
** Phase 1 - Check Blocks and Sizes  
528627 BAD I=66  
** Phase 2 - Check Pathnames  
DUP/BAD I=66 OWNER=root MODE=100755  
SIZE = 78409 MTIME = jul 16 18:45 1997
```

```
FILE=/usr/src/sys/ux
REMOVE ? y
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Count
DUP/BAD I=66 OWNER=root MODE=100755
SIZE = 78409 MTIME = jul 16 18:45 1997
FILE=/usr/src/sys/ux
CLEAR ? y
UNREF FILE I = 36 OWNER = root MODE = 100600
SIZE = 0 MTIME = jul 17 09:40 1997
RECONNECT ? y
** Phase 5 - Check Free List
157 BLK(S) MISSING
BAD FREE LIST
SALVAGE ? y
** Phase 6 - Salvage Free List
302 files 5833 blocks 371 free
```

Обратите внимание на появление шестой фазы, которой не было при нормальном выполнении команды. На этой фазе уточняется и восстанавливается список свободных блоков. После выполнения команды `fsck`, связанного с «починкой» файловой системы, может появиться сообщение:

```
***** BOOT UNIX (NO SYNC!) *****,
```

требующее перезагрузки системы без выполнения команды `sync`.

Если этого не сделать, работа по восстановлению списка свободных блоков будет утрачена, так как копии управляющих таблиц и буфера в оперативной памяти остались старыми. Для их обновления требуется перезагрузка без выгрузки буферов на диск командой `sync`.

Необходимым условием правильной работы команды `fsck` является наличие пустого каталога `/lost+found`, расположенного в корневом каталоге. Если при выполнении команды `fsck` будут найдены каталоги, на которые никто не ссылается в проверяемой файловой системе, они будут подключены в каталог `/lost+found` для дальнейшего изучения их принадлежности.

Контроль использования дисковой памяти. Регулярное выполнение команды `du` (disk usage) позволяет выявить пользователей, захвативших слишком много дисковой памяти. Команда печатает число блоков, занятых каждым файлом и каталогом в дереве, указанном параметром команды (именем каталога).

Пример:

```
du /
```

В результате исполнения этой команды будут выведены объемы всех файлов и каталогов.

Использование команды `find` помогает выявить долго не используемые файлы большого объема. Например, для того чтобы выявить все файлы, к которым не обращались ни по записи, ни по чтению последние 90 дней, следует ввести команду:

```
find / -m time+90 -a tirae+90 -print
```

С помощью команды `find` администратор может также найти файлы, представляющие опасность для операционной системы или бесполезно занимающие пространство на диске, даже если место расположения этих файлов в иерархии каталогов неизвестно, например:

```
find / -name danger -print
```

При этом поиск производится начиная с корневого каталога `/`. Ключ `-name` указывает последующее имя файла `danger`, а ключ `-print` предписывает вывод полного имени файла `danger` на экран.

Команда `df` (`disk free`) показывает число всех свободных блоков или, если она явно указана, — определенной файловой системы.

Следует помнить, что учетные файлы (типа `/usr/lib/cronlog`) могут расти неограниченно и требуют периодической чистки или сброса в архив.

Работа с руководствами для пользователя

Тексты руководств для пользователей находятся в различных подкаталогах `man?` каталога `/usr/man`, где вопросительный знак — метасимвол, принимающий значения от 1 до 8, в соответствии с нумерацией руководств по системе. Файлы этих подкаталогов содержат исходные тексты отдельных руководств. Например, подкаталог `man1`, содержащий файлы текстов команд, может иметь следующий вид (фрагмент):

```
> ls /usr/man/man1
...
```

```
cpio.1 grep.1 mknod.1 ren.1 test.1
...
```

Исходные тексты руководств хранятся в этих файлах в виде, подготовленном для команды форматирования `nroff` (`newrun-off`). На выходе команды форматирования появится текст в формате, пригодном для выдачи на печать или терминал. Во многих системах текст руководства в выходном формате создается и запоминается в файлах каталога `/usr/man/cat?/*`, где `?` и `*` — метасимволы в обычном смысле в процессе выполнения команды `catman`.

Выдача руководства на терминал или печать выполняется командой `man`, которая в соответствии с указанным аргументом ищет вначале текст руководства в выходном формате и выдает его на стандартный вывод. В противном случае, она ищет исходный текст и вызывает команду `nroff`, результат работы которой поступает на стандартный вывод. Добавление новых руководств требует знания форматов для команды `nroff`, работающей с пакетом `man`-макроопределений.

Структура информации о функциях Unix. Руководство Unix Reference Manual содержит восемь разделов:

1. Commands (команды).
2. System calls (системные вызовы).
3. Subroutines (подпрограммы).
4. Special files (спецфайлы).
5. File format and conversion (формат файлов и соглашения).
6. Games (игры).
7. Macro packages and language Conventions (макропакеты и языковые соглашения).
8. Administrator commands and procedures (команды и процедуры администратора).

Описание команды состоит из следующих разделов (см. рис. 3.36):

```
NAME (имя и функция);
SYNOPSIS (синтаксис);
DESCRIPTION (описание функции);
FILE (используемые файлы);
SEE ALSO (смежные команды);
DIAGNOSTIC (реакция на ошибки);
BUGS (замеченные некорректности).
```

Internet-возможности Unix

Система архивов FTP. FTP-архивы — это распределенный депозитарий разнообразных данных, накопленных в сети Internet за последние десятилетия. Любой пользователь может воспользоваться анонимным доступом к этому хранилищу и скопировать интересующие его материалы. Только объем программного обеспечения в архивах FTP составляет терабайты информации; кроме того, в FTP-архивах можно найти стандарты Internet (RFC), пресс-релизы, книги по различным отраслям знаний, главным образом по компьютерной проблематике, и многое другое.

Информация в FTP-архивах разделена на три категории (рис. 3.10):

- защищенная информация, режим доступа к которой определяется ее владельцами и разрешается по специальному соглашению с потребителем. К этому виду ресурсов относятся коммерческие архивы (например, коммерческие версии программ в архивах `ftp.microsoft.com` или `ftp.bsdi.com`), закрытые национальные и международные некоммерческие ресурсы (например, работы по международным проектам CERN или IAEA), частная некоммерческая информация со специальными режимами доступа (например, благотворительные фонды);
- информационные ресурсы ограниченного использования, к которым относятся, например, программы класса shareware. В данный класс могут входить ресурсы ограниченного времени использования или ограниченного времени действия, т. е. потребитель может использовать текущую версию на свой страх и риск, но никто не будет оказывать ему поддержку;

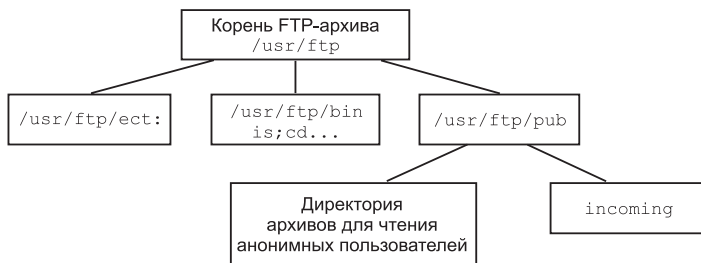


Рис. 3.10. Структура FTP-архива

- свободно распространяемые информационные ресурсы, или freeware, если речь идет о программном обеспечении. К этим ресурсам относится все, что можно свободно получить по сети без специальной регистрации. Это может быть документация, программы или что-либо еще. Наиболее известными свободно распространяемыми программами являются программы проекта GNU Free Software Foundation. Следует отметить, что свободно распространяемое программное обеспечение не имеет сертификата качества, но, как правило, его разработчики открыты для обмена опытом.

Архивы используют для решения разных задач, однако наиболее популярными в сети являются *свободно доступные* или такие архивы, доступ к которым разрешен по анонимному идентификатору пользователя. Таким образом, эти архивы можно использовать в качестве:

- коллекции свободно распространяемого программного обеспечения;
- коллекции программ для бета-тестирования;
- коллекции нормативных и регламентных документов и т. п.

FTP-архив может служить и в качестве архива коммерческого программного обеспечения, которое используется в компании, только в этом случае такой архив не должен разрешать анонимного доступа к хранящимся в нем ресурсам.

Технология FTP была разработана в рамках проекта ARPA и предназначена для обмена большими объемами информации между машинами с различной архитектурой. Стержень технологии составляет FTP-протокол. Поскольку главным в проекте было обеспечение надежной передачи, с современной точки зрения FTP-протокол кажется перегруженным излишними редко используемыми возможностями.

Это один из старейших протоколов в Internet; обмен данными в FTP проходит по TCP-каналу и построен по технологии «клиент — сервер». Практически для любой платформы и операционной среды существуют как серверы, так и клиенты. Ниже кратко описываются стандартные сервер и клиент Unix-подобных систем.

Программное обеспечение доступа к FTP-архивам. Для работы с FTP-архивами необходимо следующее программное обеспечение: сервер, клиент и поисковая программа. Сервер обеспечивает доступ к ресурсам архива из любой точки сети, клиент — до-

ступ пользователя к любому архиву в сети, а поисковая система — навигацию во всем множестве архивов сети.

Сервер протокола — программа ftpd. Команда ftpd предназначена для обслуживания запросов на обмен информацией по протоколу FTP. Сервер обычно стартует в момент загрузки компьютера. Входит в состав ОС Unix. Каждый сервер имеет свое описание команд, которое можно получить по команде help.

Возможен вход в архив по идентификатору пользователя anonymous или ftp. В этом случае сервер принимает меры по ограничению доступа к ресурсам компьютера для данного пользователя. Обычно для таких пользователей создается специальная директория ftp, в которой размещают каталоги bin, etc и pub. В каталоге bin размещаются команды, разрешенные для использования, а в каталоге pub — собственно файлы. Каталог etc закрыт для просмотра пользователем и там размещены файлы идентификации пользователей.

Программа обмена файлами ftp — это командный интерфейс пользователя при обмене файлами по одноименному протоколу. Программа входит в состав ОС Unix, устанавливает канал управления с удаленным сервером и ожидает команд пользователя.

Если программа ftp работает с пользователем и ожидает его команд, то на экране отображается приглашение ftp>.

Вот некоторые команды данного режима:

open — по этой команде открывается сеанс работы с удаленным сервером:

```
ftp>open polyn.net.kiae.su;
```

После выдачи команды последуют запросы идентификации пользователя.

user — регистрация пользователя

```
ftp> user anonymous;
```

В данном случае пользователь не имеет особых прав доступа на удаленном сервере и поэтому регистрируется как аноним. В ответ на запрос идентификации пользователю следует ввести свой почтовый адрес.

cd и ls (dir) — назначение этих команд достаточно очевидно — навигация по дереву файловой системы и просмотр содержания каталогов. Так как в процессе приема-передачи участвуют

две машины, то, кроме навигации, в удаленной файловой системе нужна еще навигация в локальной файловой системе. Для этой цели служит команда `lcd` (локальная `cd`).

Кроме того, пользователь может выдать и любую команду локальной оболочки, если предварит ее символом «!»:

```
ftp> !pwd,
```

По этой команде будет выдано имя текущей директории на локальной машине.

Команды `get`, `put`, `mget`, `mput` и `bin` (прием/передача данных). По командам `get` и `put` можно принять или передать один файл:

```
ftp> get readme.txt;
```

Команды `mget`, `mput` предназначены для приема/передачи группы файлов:

```
ftp> mget *.gz;
```

`prompt` — обычно при передаче групп файлов для каждого файла запрашивается подтверждение. Для того чтобы избежать этого перед приемом/передачей, следует выдать команду `prompt`. Последняя переключает режим запроса подтверждения и при повторном использовании этой команды состояние запроса подтверждения восстанавливается;

`bin` — после выдачи этой команды по умолчанию данные будут передаваться в режиме передачи двоичных данных, что весьма существенно, так как в режиме ASCII нельзя передать программы и архивированные данные. Часто бывает полезно включить режим `bin` для символьных данных с произвольной длиной строки, например файлов `postscript` (*.ps);

`quit` — команда выхода из ftp.

Управление устройствами

Управление внешними устройствами — это одна из важнейших функций любой операционной системы. Система должна обеспечивать эффективный и удобный доступ к периферийным устройствам, а также обеспечивать возможность унифицированной разработки программного обеспечения для вновь подключаемых внешних устройств. Рассмотрим, как эта проблема решается в ОС Unix.

Устройство как специальный файл. Для доступа к внешним устройствам в ОС Unix используется универсальная абстракция файла. Помимо настоящих файлов (обычных файлов или каталогов), которые реально занимают память на магнитных дисках, файловая система содержит так называемые специальные файлы, для которых, как и для настоящих файлов, отводятся отдельные *i*-узлы, но которым на самом деле соответствуют внешние устройства. Это решение позволяет естественным образом работать в одном и том же интерфейсе с любым файлом или внешним устройством.

Драйверы устройств. Очевидно, что простое объявление внешнего устройства специальным файлом не даст возможности работать с этим устройством, если не создан и соответствующим образом не подключен к системе специальный программный код, соответствующий специфике данного устройства. Как и в большинстве современных операционных систем, такого рода программный код в ОС Unix называется драйвером устройства (в этом контексте слово драйвер лучше всего понимать в значении «управляющий»).

В ОС Unix различаются символьные, блочные и потоковые драйверы. Символьные драйверы являются простейшими в ОС Unix и предназначены для обслуживания устройств, которые реально ориентированы на прием или выдачу произвольных последовательностей байтов (например, простой принтер или устройство ввода с перфоленты). Такие драйверы используют минимальный набор стандартных функций ядра Unix, которые главным образом заключаются в возможности взять данные из виртуального пространства пользовательского процесса и/или поместить данные в такое виртуальное пространство.

Блочные драйверы более сложны. Они работают с использованием возможностей системной буферизации блочных обменов ядра ОС Unix. В число функций такого драйвера входит включение соответствующего блока данных в систему буферов ядра ОС Unix и/или взятие содержимого буферной области в случае необходимости.

Наконец, наиболее сложной организацией отличаются потоковые драйверы. Фактически, такой драйвер представляет собой конвейер модулей, обеспечивающий многоступенчатую обработку запросов пользователя. Потоковые драйверы в среде ОС Unix в основном предназначены для реализации доступа к сетевым устройствам, которые должны работать в соответствии с многоуровневыми сетевыми протоколами.

Управление процессами и потоками (threads)

В операционной системе Unix традиционно поддерживается классическая схема мультипрограммирования. Система поддерживает возможность параллельного (или квазипараллельного в случае наличия только одного аппаратного процессора) выполнения нескольких пользовательских программ. Каждому такому выполнению соответствует процесс операционной системы. Каждый процесс выполняется в собственной виртуальной памяти, и, тем самым, процессы защищены один от другого, т. е. один процесс не в состоянии неконтролируемым образом прочитать что-либо из памяти другого процесса или записать в нее. Однако контролируемые взаимодействия процессов допускаются системой, в том числе за счет возможности разделения одного сегмента памяти между виртуальной памятью нескольких процессов.

Ядро системы предоставляет возможности (набор системных вызовов) для порождения новых процессов, отслеживания окончания порожденных процессов и т. д. С другой стороны, в ОС Unix ядро системы — это полностью пассивный набор программ и данных. Любая программа ядра может начать работать только по инициативе некоторого пользовательского процесса (при выполнении системного вызова) либо по причине внутреннего или внешнего прерывания (примером внутреннего прерывания может быть прерывание из-за отсутствия в основной памяти требуемой страницы виртуальной памяти пользовательского процесса; примером внешнего прерывания является любое прерывание процессора по инициативе внешнего устройства). В любом случае считается, что выполняется ядерная часть обратившегося или прерванного процесса, т. е. ядро всегда работает в контексте некоторого процесса.

В связи с распространением так называемых симметричных мультипроцессорных архитектур компьютеров (Symmetric Multi-processor Architectures — SMP) в ОС Unix был внедрен механизм легковесных процессов (light-weight processes), или нитей, или потоков управления (threads). Можно сказать, что нить — это процесс, выполняющийся в виртуальной памяти, используемой совместно с другими нитями одного и того же «тяжеловесного» (т. е. обладающего отдельной виртуальной памятью) процесса.

Пользовательская и ядерная составляющие процессов. Каждому процессу соответствует контекст, в котором он выполняется.

Этот контекст включает содержимое пользовательского адресного пространства — пользовательский контекст (т. е. содержимое сегментов программного кода, данных, стека, разделяемых сегментов и сегментов файлов, отображаемых в виртуальную память), содержимое аппаратных регистров — регистровый контекст (регистр счетчика команд, регистр состояния процессора, регистр указателя стека и регистры общего назначения), а также структуры данных ядра (контекст системного уровня), связанные с этим процессом. Контекст процесса системного уровня в ОС UNIX состоит из «статической» и «динамических» частей. Для каждого процесса имеется одна статическая часть контекста системного уровня и переменное число динамических частей.

Статическая часть контекста процесса системного уровня включает описатель процесса, т. е. элемент таблицы описателей существующих в системе процессов. Описатель процесса содержит, в частности, следующую информацию:

- состояние процесса;
- физический адрес в основной или внешней памяти *u-области процесса*;
- идентификаторы пользователя, от имени которого запущен процесс;
- идентификатор процесса;
- прочую информацию, связанную с управлением процессом.

U-область (*u-area*) — индивидуальная для каждого процесса область пространства ядра, обладающая тем свойством, что хотя *u-область* каждого процесса располагается в отдельном месте физической памяти, *u-области* всех процессов имеют один и тот же виртуальный адрес в адресном пространстве ядра. Именно это означает, что какая бы программа ядра ни выполнялась, она всегда выполняется как ядерная часть некоторого пользовательского процесса и именно того процесса, *u-область* которого является «видимой» для ядра в данный момент времени. U-область процесса содержит:

- указатель на описатель процесса;
- идентификаторы пользователя;
- счетчик времени, в течение которого процесс реально выполнялся (т. е. занимал процессор) в режиме пользователя и режиме ядра;
- параметры системного вызова;
- результаты системного вызова;

- таблицу дескрипторов открытых файлов;
- предельные размеры адресного пространства процесса;
- предельные размеры файла, в который процесс может писать, и т. д.

Динамическая часть контекста процесса — это один или несколько стеков, которые используются процессом при его выполнении в режиме ядра. Число ядерных стеков процесса соответствует числу уровней прерывания, поддерживаемых конкретной аппаратурой.

3.4. Операционная система Linux

Linux — свободно распространяемая версия Unix, первоначально была разработана Линусом Торвальдсом (Linus Torvalds). Linux была создана с помощью многих Unix-программистов и энтузиастов из Internet, тех, кто имел достаточно навыков и способностей развивать систему. Ядро Linux не использует коды AT&T или какого-либо другого частного источника, и большинство программ Linux разработаны в рамках проекта GNU из Free Software Foundation в Cambridge, Massachusetts, но в него внесли свой вклад также программисты всего мира.

Системные характеристики

Linux — полная многозадачная многопользовательская операционная система (точно так же, как и другие версии Unix).

Другие специфические внутренние черты Linux включают контроль работ по стандарту POSIX (используемый оболочками, такими, как `cs`h и `ba`sh), псевдотерминалы (`pty`), поддержку национальных и стандартных клавиатур динамически загружаемыми драйверами клавиатур.

Ядро может эмулировать команды 387-FPU, так что компьютеры без сопроцессора могут выполнять программы с плавающей точкой.

Ядро Linux создано с учетом специального защищенного режима для процессоров Intel. В частности, Linux использует парадигму описания памяти в защищенном режиме и другие свойства процессоров.

Для увеличения объема доступной памяти Linux осуществляет также разбиение диска на страницы: т. е. на диске может быть выделено «пространство для свопинга» (swar space). Здесь swar space не совсем подходящий термин, поскольку в Linux в область свопинга выгружается не весь процесс, а только отдельные его части, в которых временно нет необходимости. Когда системе нужно больше физической памяти, то она с помощью свопинга выводит неактивные страницы на диск.

Ядро также поддерживает универсальный пул памяти для пользовательских программ и дискового кэша. При этом для кэша может использоваться вся память, и наоборот, кэш уменьшается при работе больших программ.

Выполняемые программы используют динамически связываемые библиотеки, т. е. выполняемые программы могут совместно использовать библиотечную программу, представленную одним физическим файлом на диске (иначе, чем это реализовано в механизме разделяемых библиотек SunOS). Это позволяет выполняемым файлам занимать меньше места на диске, особенно тем, которые многократно используют библиотечные функции.

Совместимость. Linux представляет собой комбинацию BSD Unix и System V Release 4 Unix. Linux достаточно хорошо совместима с рядом стандартов на уровне исходных текстов, включая IEEE POSIX.1, System V и BSD.

Графический интерфейс. В Linux для построения графических интерфейсов (GUI) применяется система библиотек X Window. На этой базе разработаны различные программы управления интерфейсами пользователя, например, KDE и GNOME, которые позволяют при желании сделать Desktop (рабочий стол) Linux похожим на Desktop Windows (см. рис. 3.14).

Поддерживаемые файловые системы. Linux позволяет работать с различными типами файловых систем (табл. 3.6). Некоторые из них, такие как EXT2/3, ReiserFS, JFS и XFS, считаются базовыми для Linux. Поддержка FAT (файловая система MS DOS) позволяет непосредственно обращаться к файлам MS DOS/Windows на жестком диске.

Сетевые возможности. Linux обеспечивает полный набор протоколов TCP/IP для работы в сети Internet. Поддерживается весь спектр клиентов и услуг TCP/IP, таких как FTP, TELNET, NNTP и SMTP. Имеется возможность получить доступ к сети Internet без установки сетевого адаптера, посредством установки

Таблица 3.6. Файловые системы, поддерживаемые Linux

ФС	Пояснения
EXT2/3	Улучшенные версии EXTFS (Extended File System), предложенной в 1992 г. для Linux 0.96c
ReiserFS	Разработана Гансом Райзером (Gans Reiser) и рабочей группой NameSys как альтернатива EXT2. ФС Reiser4 использует принципы «plug-ins»
JFS	Journaling File System разработана фирмой IBM для 64-битовых систем
XFS	Разработана SGI (Silicon Graphic) для 64-битовых систем
CRAMFS	Файловая система для сжатой информации ROM (доступ только на чтение)
HPFS	High Performance File System — стандартная ФС IBM OS/2 (поддерживается только чтение)
ISO9660	Стандартная файловая система для CD-ROM
Minix	ФС была предложена одной из первых для Linux, в настоящее время используется на гибких дисках
FAT	Основная система DOS, поддерживаемая многими современными ОС
NCPFS	ФС Novell для использования в сетях
NFS	Network File System — доступ к данным, установленным на различных машинах в сети
SMBFS	Server Message Block используется продуктами типа Windows для доступа к файлам через сеть
SYSV	Используется версиями SCO Unix, Xenix и Coherent (коммерческий вариант Unix для ПЭВМ)
UFS	Используется системами BSD, SunOS и NeXTstep, поддерживается только считывание данных
UMSDOS	Эмуляция Unix в MS-DOS — надстройка над обычной FAT, обеспечивающая Unix-подобные возможности (связь, разделение доступа, поддержка длинных имен и пр.) путем добавления вспомогательных системных файлов
VFAT	Виртуальная FAT — расширение FAT (поддержка длинных имен файлов)
NTFS	Файловая система Windows NT/2000, в режиме только чтения

модема и протокола PPP, а также при наличии многопортовых модемов или последовательных плат — Linux обеспечивает эффективный и надежный шлюз PPP для удаленных пользователей, звонящих по коммутируемой линии.

Небольшие требования к вычислительным ресурсам. В Linux объединены мощь и гибкость рабочей Unix-станции, возмож-

ность использования полного набора приложений Internet и полнофункциональный графический интерфейс при незначительных требованиях к вычислительным ресурсам.



Версии ОС. Существуют различные версии и дистрибутивы Linux — Mandrake, Red Hat, Alt и пр. В качестве примера для иллюстраций в настоящем пособии используется версия SuSE Linux 11.0. Пиктограмма «хамелеон» удачно иллюстрирует вариативные возможности оболочки.

Оконная система X (X window system)

Для нормальной организации работы пользовательских программ с графическими терминалами требуется наличие базового слоя программного обеспечения со следующими возможностями:

- скрывающего аппаратные особенности терминала;
- обеспечивающего создание окон на экране терминала, управление этими окнами и работу с ними со стороны пользовательской программы;
- дающего возможность пользовательской программе реагировать на события, происходящие в соответствующем окне (ввод с клавиатуры, движение курсора, нажатие клавиш мыши и т. д.).

Такой базовый слой графического программного обеспечения принято называть *о к о н н о й с и с т е м о й*.

Пользователями Unix предпринимался ряд попыток создания оконных систем, и большинство из них успешно использовалось практически (например, оконная система NeWS компании Sun Microsystems, интерфейс которой основывался на использовании языка PostScript). Однако ни одна из этих систем не выходила за пределы ведомственного использования, что ограничивало мобильность программ, обладающих графическим интерфейсом. Успеха удалось добиться группе молодых исследователей и программистов из Массачусетского технологического института, которые создали оконную систему под кратким названием X (по-английски ее называют не X Window, а X window system, т. е. «оконная система X»). Система X, дополнительные библиотеки, а также ряд готовых интерфейсных средств распространяются MIT бесплатно (относясь к категории public domain). В то же

время сегодня именно оконная система X является базовым механизмом организации графических интерфейсов пользователя в большинстве Unix-систем.

Общая организация X Window. Организация системы X соответствует общей идеологии ОС Unix, которая является традиционно сетевой ОС. Девиз Б. Джоя и всей компании Sun Microsystems «The Network is the Computer» (Сеть — это компьютер) в полной мере относится к направлению ОС Unix в целом. Популярная ныне архитектура организации программно-аппаратных средств «клиент — сервер» всегда была совершенно естественной в мире Unix. Специализация и разделение функций в сети — это и значит, что для пользователя компьютер и сеть неразличимы.

Оконная система X предоставляет требуемые возможности. На стороне пользовательского терминала находится сервер системы X, обеспечивающий единообразное управление графическим терминалом вне зависимости от его специфических аппаратных характеристик. В других компьютерах сети (которые, фактически, являются серверами с точки зрения организации вычислительного процесса) установлены клиентские части системы X, создающие у выполняемой программы впечатление, что она взаимодействует с локальным терминалом, а на самом деле поддерживающие точно специфицированный протокол взаимодействий с сервером системы X.

Клиентская и серверная части оконной системы X, хотя в целом и соответствуют идеологии архитектуры «клиент — сервер», но обладают тем своеобразием, что серверная часть системы находится вблизи пользователя (т. е. основного клиента вычислительной сети), а клиентская часть системы базируется на мощных серверах сети. Конечно, система X обладает достаточной гибкостью, чтобы допустить расположение серверной и клиентской частей системы в одном компьютере, в разных компьютерах одной локальной сети и удаленных компьютерах, входящих в состав территориально распределенной сети. В зависимости от конфигурации системы X-сервер может обслуживать один или несколько графических экранов, клавиатуру и мышь, реально представляя собой процесс, группу процессов или выделенное компьютерное устройство (X-терминал).

Основой взаимодействия между клиентом и сервером оконной системы X является так называемый X-протокол, представляющий собой точную спецификацию допустимых запросов от

клиента к серверу и допустимых ответов сервера к клиенту и обладающий следующими характеристиками:

- при использовании X-протокола обработка взаимодействий клиента и сервера ведется единообразно независимо от того, основана она на внутренних механизмах IPC или на реальных сетевых обменах. Это позволяет добиться прозрачности сетевой среды как с точки зрения конечного пользователя, так и с точки зрения разработчика прикладных программ;
- за счет наличия строгой и не зависящей от окружения спецификации X-протокол может быть реализован на различных языках в различных операционных средах;
- X-протокол может быть реализован на основе любого надежно поддерживаемого потока байтов (обеспечиваемого внутренними механизмами IPC или внешними сетевыми механизмами). Многие из пригодных механизмов являются стандартными и реализованы в большинстве архитектур.

Для большинства (хотя и не для всех) приложений X-протокол не порождает существенных задержек при работе с графическими терминалами. Обычно задержки вызываются, скорее, временными потребностями самих терминалов, а не расходами на протокольные взаимодействия клиента и сервера.

Одним из клиентов оконной системы обычно является так называемый «оконный менеджер» (window manager). Это специально выделенный клиент оконной системы, обладающий полномочиями на управление расположением окон на экране терминала. Некоторые из возможностей X-протокола (связанные, например, с перемещением окон) доступны только клиентам с полномочиями оконного менеджера. Во всем остальном оконный менеджер является обычным клиентом.

Базовые библиотеки. Любая прикладная программа, которой требуется взаимодействовать с X-сервером, могла бы работать с ним, обмениваясь сообщениями в соответствии с X-протоколом. Однако, это неудобно, поскольку для выполнения любого, самого простого действия с терминалом клиенту требуется обмениваться несколькими сообщениями с X-сервером, причем для наиболее распространенных действий последовательность таких сообщений предопределена. Библиотека Си-функций, которая поставляется вместе с оконной системой X и облегчает взаимодействие Си-программы с X-сервером в соответствии с X-протоколом, называется XLib. Сам X-протокол достаточно компактен,

поскольку в нем специфицированы мелкие сообщения, которые, как правило, можно реально использовать только в некоторых комбинациях. XLib — довольно большая библиотека, поскольку каждая функция библиотеки XLib основана на использовании нескольких протокольных сообщений.

Ключи при запуске программ и их интерпретация. В отличие от программ командной строки, у которых бывают однобуквенные ключи, которые можно группировать вместе (`ls -laR`), и длинные ключи, начинающиеся с двойного минуса (`ls -help`), у программ под X бывают только длинные ключи, причем начинаются они всегда с одиночного символа «-».

Разбор командной строки выполняется не самой программой, а специальной библиотечной функцией, и существует несколько ключей, поддерживаемых всеми программами (табл. 3.7).

Таблица 3.7. Стандартные ключи X-программ

Ключ	Назначение
<code>-help</code>	Выдать краткую справку по ключам программы
<code>-display дисплей</code>	Запустить программу на указанный дисплей
<code>-fg цвет</code> <code>-foreground цвет</code>	Цвет букв («цвет переднего плана»)
<code>-bg цвет</code> <code>-background цвет</code>	Цвет фона
<code>-fn шрифт</code> <code>-font шрифт</code>	Основной шрифт
<code>-geometry геометрия</code>	Сделать окно указанного размера и/или поместить его в указанном месте на экране
<code>-iconic</code>	Запустить программу сразу «свернутой» в пиктограмму

Управление цветами. Поскольку X-Window работает с дисплеями, у которых цвет кодируется набором RGB (Red, Green, Blue), то использует именно такое представление. Цвета можно указывать одним из двух способов: либо триплетом RGB в шестнадцатеричном виде, либо по имени.

При первом способе цвет указывается как `#RRGGBB` — символ «решетка» и затем шесть шестнадцатеричных цифр — по две цифры на красный, зеленый и синий.

Например, #FFFFFF — белый, #000000 — черный, #CC00CC — темно-фиолетовый, а #92FF41 — желто-зеленый.

Во втором случае указывается английское название — white, black, red, darkblue, maroon и т. д.

X «знает» по именам несколько сотен цветов. В их число входят «стандартные» цвета типа white, blue, green, их подвиды типа lightblue, darkblue, градации яркости — SteelBlue1 ... SteelBlue4, а также «экзотика» типа moccasin и PowderBlue. Многие из этих цветов на глаз практически неразличимы (а некоторые и просто фигурируют под разными названиями, например, cyan и cyan1).

Посмотреть список цветов (и их определений) можно командой showrgb. Она печатает R-, G- и B-компоненты цвета (в десятичной системе) и его имя.

Ключи -fg и -bg позволяют указывать программам, какие цвета использовать для букв и для фона.

Управление шрифтами в X Window. Для сравнения напомним, что в Windows для однозначного указания шрифта служат три параметра — гарнитура, начертание и размер. Например, Arial Bold 12pt (Гарнитура Arial, полужирный, размер 12 пунктов).

В X Window имя шрифта состоит из 14 компонентов, разделенных дефисами, например:

```
-b&h-lucida-bold-r-normal-sans-12-120-75-75-p-79-iso8859-1
```

Общий формат такой:

```
-fndry-fmly-wght-slant-sWdth-astyl-pxlsz-ptSz-resx-resy-  
spc-avgWdth-rgstry-encdng
```

Поля fndry и fmly — это фирма-создатель и гарнитура шрифта — аналог *гарнитуры* в Windows (может одновременно использоваться, например, два разных шрифта Times от разных фирм).

Поля wght, slant и sWdth — это соответственно «жирность» (вес) шрифта, наклонность и ширина — аналог «начертания» в Windows.

Поле pxlsz указывает размер шрифта в пунктах.

Поле spc определяет вид шрифта — постоянной ширины (m — monospace) или пропорциональный (p — proportional).

Пара полей rgstry и encdng задает кодировку шрифта, например, iso8859-1 или koi8-r.

Обычно не требуется указывать шрифт с полной конкретностью, т. е. перечислять все его 14 параметров, а нужен, к примеру, «постоянной ширины, прямой, 12 пунктов». В этом случае достаточно воспользоваться шаблоном — указать лишь требуемые параметры, а в остальных позициях — «*»:

```
-b&h-*-medium-r-*-14-*-**-*-*-*.*
```

X Window автоматически подберет первый подходящий шрифт (рис. 3.11).

Хотя такой «детальный» способ очень удобен для программ, человеку запоминать и набирать такие длинные имена неудобно и поэтому для многих шрифтов есть «псевдонимы» (aliases) — короткие имена.

К примеру, под названием

```
lucidasans-italic-12
```

подразумевается шрифт

```
-b&h-lucida-medium-i-normal-sans-12-120-75-75-p-71-iso8859-1.
```

Программы для выбора и просмотра шрифтов: *xfontsel* и *xfid*.

Для подбора шрифта служит программа *xfontsel* (X Font Selector) — рис. 3.11.

Она отображает меню в виде формата шрифта, в котором можно выбирать нужные параметры, текущий выбранный шаблон, и как выглядит первый подходящий к нему шрифт.

Поскольку шрифты есть не для всех комбинаций параметров, то по мере выбора параметров из меню некоторые пункты в других меню становятся «недоступными», например, шрифты гарнитуры Times есть только в кодировках iso8859 и koi8, а в jis — нет.

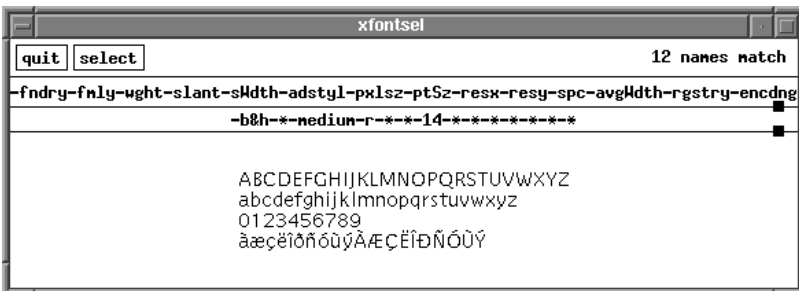


Рис. 3.11. Окно программы *xfontsel*

Подобрав шрифт, можно нажать кнопку `select` — при этом название будет скопировано в буфер обмена, и его затем можно вставить, например, в `xterm` средней кнопкой мыши.

Для просмотра всех символов шрифта (а не только тех, что отображает в своем примере `xfontsel`, служит программа `xfd` (X Font Displayer) — рис. 3.12.

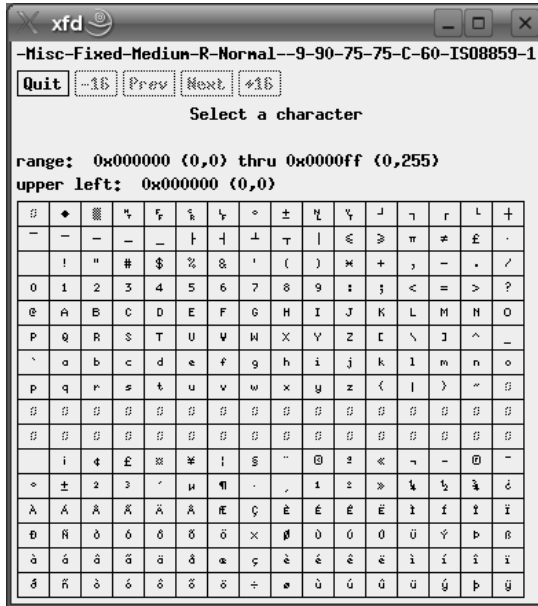


Рис. 3.12. Окно программы `xfd`, запущенной командой `xfd -fn 9x12`

Ей надо указывать требуемый шрифт (или шаблон) ключом `-fn`. При этом имя шрифта следует заключать в одинарные апострофы, поскольку оно обычно содержит символы, имеющие специальное значение для интерпретатора командной строки (shell), такие, как «*» и «&».

Геометрия окон при запуске программ. Координаты и размеры окон от запуска к запуску программы не запоминаются, но их можно указать при запуске. Для этого служит ключ `-geometry`. Под «геометрией» окна понимается его размер и позиция на экране.

У ключа `-geometry` есть три варианта:

- `-geometry Размер`;
- `-geometry Позиция`;

- `-geometry` *РазмерПозиция*.

Размер указывается как *ШиринаxВысота* (между размерами — латинская буква *x*). В зависимости от программы ширина и высота указываются или в пикселях (в большинстве программ) или в символах (например, в *Xterm*). Причем указывается размер внутренней части окна, без учета обрамления (которое зависит от window manager и может быть каким угодно).

Позиция указывается в пикселях от края экрана. Обычно она выглядит как `+X+Y`. В этом случае левый край окна будет в *X* пикселях от левого края экрана, а верхний край — в *Y* пикселях от верхнего края экрана.

Но иногда нужно поместить окно, например, вплотную к правому краю. В этом случае надо в горизонтальной координате вместо «+» указать «-». При этом окно будет расположено так, что его правый край окажется в *X* пикселях от правого края экрана.

Аналогично вертикальную позицию можно также указывать от нижнего края экрана. Несколько примеров приведено на рис. 3.13.

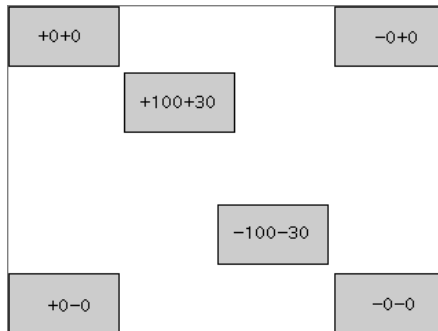


Рис. 3.13. Координаты окон на экране

Несколько примеров указания геометрии:

```
xterm -fn 6x10 -geometry 80x24+30+200 &  
xclock -geometry 48x48-0+0 &  
xload -geometry 48x48-96+0 &  
xbiff -geometry 48x48-48+0 &
```

Здесь *xterm* помещается приблизительно в левой средней части экрана, а часы, индикаторы загрузки и писем — в правый верхний угол.

Интерактивная среда K Desktop Environment (KDE)



KDE — интерактивная графическая рабочая среда, набор программ, технологий и документации, которые призваны облегчить работу пользователей персональных компьютеров. KDE предназначена для работы на компьютерах под управлением Unix.

KDE возникла как ответ на потребность в удобной в использовании рабочей среде для рабочих станций под Unix, аналогичной уже существующим системам на базе Mac OS или Windows. Основные инструменты для достижения этой цели — это улучшенные средства межпрограммных связей, повторное использование компонентов, технология «drag and drop», единый внешний вид и многое другое.

Без Unix не было бы и Internet, по крайней мере, в его теперешнем представлении. Однако до недавнего времени Unix не отвечала запросам индивидуального пользователя. Этот факт особенно негативен, поскольку существуют такие варианты Unix (Linux и FreeBSD, NetBSD и др.), которые свободно доступны в Internet и славятся исключительным качеством и стабильностью.

Стабильность, масштабируемость и открытость — качества, которые сделали Unix основным выбором профессионалов в области информационных технологий в течение многих лет. Все эти качества лежат в основе KDE, и, в свою очередь, она сама предлагает удобство в использовании и дружелюбность к пользователю. KDE была и остается лидером в том, чтобы наделить всеми эти привлекательными чертами Unix — систему, которая всегда была платформой номер один для серверных применений и в научных учреждениях, но не привлекала рядовых пользователей.

KDE — интерактивная рабочая среда. В сочетании со свободными версиями Unix KDE создает открытую и свободную от всех ограничений рабочую среду для домашнего или профессионального применения. Эта платформа доступна бесплатно, включая полные исходные тексты, что дает возможность их модификации.

KDE — среда для разработки приложений. KDE всецело фокусируется на нуждах конечного пользователя, но очевидно, что эта цель в первую очередь достигается и тем, что разработчики имеют в своем распоряжении наилучшие средства разработки. Поэтому в основу KDE положены новейшие достижения в технологии разработки программного обеспечения.

KDE — пакет офисных приложений. В настоящий момент KDE предоставляет помимо интерактивной рабочей среды еще и цельный пакет мощных офисных приложений, известный под названием KOffice. KOffice базируется на технологиях DCOP/KParts. Пакет включает в себя и дополнительные средства, такие, как клиенты электронной почты и новостей, мощный органайзер PIM (Personal Information Manager) и многое другое.

Рассмотрим далее основные элементы интерфейса и возможности KDE.

Элементы рабочего стола KDE3

Основные видимые части рабочего стола — поверхность рабочего стола с ярлыками (рис 3.14, 1) и панель (рис 3.14, 2).

Панель (Kicker). При первом запуске KDE панель располагается внизу экрана. Главным образом Kicker используется для запуска приложений посредством щелчка на пиктограмме панели или в К-Меню. На панели размещаются следующие объекты:



К-кнопка (вызова меню KDE) — одна из самых важных частей рабочего стола (рис. 3.15, 1), открывающая главное меню (К - меню), посредством которого можно запускать различные приложения, установленные в системе (рис. 3.14, 3).

Запуск программ в средах GNOME и KDE осуществляется как посредством символьческих ссылок (аналоги ярлыков в Windows), расположенных на рабочем столе, так и ещё тремя способами — из командной строки терминала (эмулятора терминала), из стартового К-меню и из Панели быстрого запуска. Создание символических ссылок выполняется точно так же, как в системах Windows.

Пиктограммы запуска приложений соответствуют часто используемым приложениям. Их можно перемещать, удалять, а также добавлять другие. Если поместить курсор мыши на пиктограмму, не щелкая по ней, появится название приложения и его описание (в том случае, если это установлено в Диалоге настроек). В данном примере можно видеть следующие пиктограммы:



— кнопка запуска (рис. 3.15 2) программной оболочки Konqueror (см. ниже, разд. 4.6) в режиме файлового менеджера;

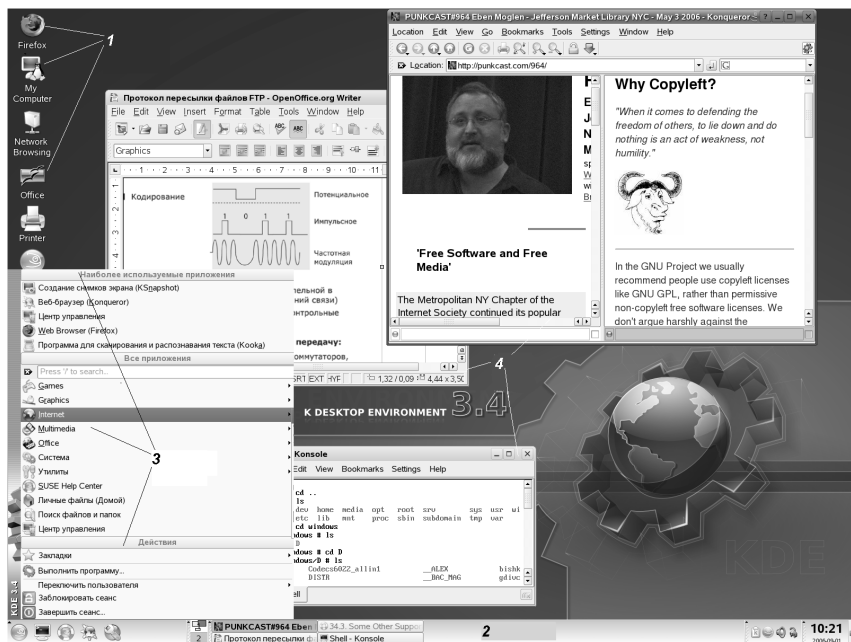


Рис. 3.14. Рабочий стол KDE 3.4:

1 — поверхность рабочего стола с ярлыками; 2 — панель; 3 — главное меню (К-меню); 4 — окна приложений (OpenOffice.org Writer, Konqueror, терминал)



Рис. 3.15. Общий вид панели рабочего стола:

1 — кнопка главного меню; 2 — менеджер файлов; 3 — терминал; 4 — центр помощи; 5 — Web-обозреватель; 6 — информационный менеджер; 7 — переключатель рабочих столов; 8 — панель задач; 9 — апплет Выйти/Запереть Окно; 10 — системный лоток; 11 — часы; 12 — кнопка сворачивания панели



— кнопка вызова (рис. 3.15, 3) эмулятора терминала Unix (тот же результат, если в основном меню выбрать Система\Терминал). При нажатии открывается окно консоли (см. рис. 3.6), в котором могут быть выполнены текстовые команды Unix — `ls`, `cat`, `less` и т. д. Используя меню Настройка, можно изменять вид и тип терминала;



— вызов (рис. 3.15, 4) Центра справки KDE (рис. 3.16);



— кнопка запуска (рис. 3.15, 5) программной оболочки Konqueror (см. ниже, разд. 4.6) в режиме Web-обозревателя;



— вызов (рис. 3.15, 6) персонального информационного менеджера Kontact;



— апплет* переключателя рабочих столов, который отображает все виртуальные рабочие столы. По умолчанию рабочих столов 4, но их может быть от 1 до 16. Во время работы можно переключаться между этими виртуальными рабочими столами, расположенными в нижней части экрана монитора, с помощью мыши. Наличие нескольких рабочих столов очень удобно, так как, открыв программы равномерно на всех рабочих столах, минимизируем количество открытых окон на каждом отдельном столе. В результате рабочие столы не перегружены приложениями, в них легче разобраться, а значит, повышается производительность работы в системе. Обмениваться результатами работы программ на разных столах можно через буфер обмена или с помощью «перетаскивания» мышью. При первом запуске KDE появляются от 2 до 4 кнопок (здесь — две, помеченные соответственно как 1 и 2, рис. 3.15, 7). Треугольная кнопка слева сверху предназначена также для переключения между столами;



— панель задач, все запущенные приложения отображаются на панели задач в виде кнопок (рис. 3.15, 8). Щелчок мыши на кнопке помещает данное приложение поверх других. Если до этого приложение находилось «наверху», оно будет свернуто в пиктограмму на панели. В зависимости от состояния окна меняется цветовая гамма кнопки (см. рис. 3.21). Нажмите на кнопку, соответствующую окну, которое вы хотите открыть.

* Апплеты — это небольшие приложения, запускаемые в Kicker. Почти все, за исключением кнопок запуска, — это апплеты. Их можно добавить из контекстного меню или из подменю Настройки панели, выбрав Добавить и пункт меню Апплет. На ручке каждого апплета изображена маленькая стрелка , и если на нее нажать, появится меню, позволяющее перемещать или удалять апплет, устанавливать его настройки (если они есть) и настройки панели Kicker.

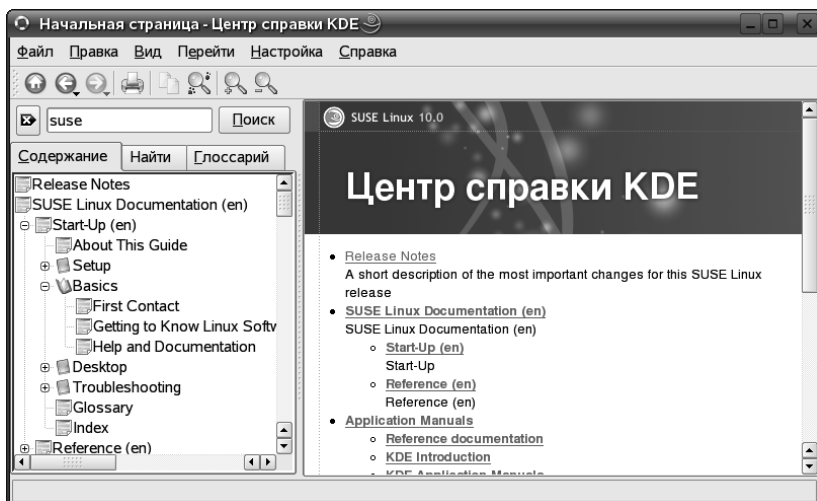


Рис. 3.16. Начальный экран центра справки KDE

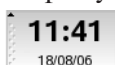
Нажав ее еще раз, вы свернете окно. При нажатии правой кнопки появится контекстное меню (см. рис. 3.23, а);



— выйти/запереть окно. Этот апплет содержит две небольшие кнопки (рис. 3.15, 9). Первая, на которой изображен замок, запирает ваш экран, если вы оставляете свою рабочую среду KDE и не хотите, чтобы кто-то другой видел или пользовался вашим рабочим столом. Вторая — завершает сеанс KDE и X-сервера;



— системный лоток, на котором могут находиться некоторые полезные приложения (рис. 3.15, 10). В KDE по умолчанию уже установлено несколько широко употребительных приложений, в нашем примере это — Klipper (управление буфером обмена), KPowerSave (регулирование потребляемой мощности), SuSE Watcher (внесение обновлений в систему), микшер звуковой системы и часы (время и дата);



— апплет часов (рис. 3.15, 11), которые, очевидно, могут принимать различный вид;



— кнопка Скрыть, которая позволяет свернуть Kicker и увеличить свободное пространство на экране (рис. 3.15, 12). Когда Kicker свернут, на экране остается только кнопка (Открыть) для того, чтобы снова развернуть Kicker.

Такой вид (рис. 3.15) принимает панель Kicker при первом запуске. В дальнейшем ее можно перенастраивать различными путями и устанавливать дополнительные апплеты.

Компоненты главного меню. В меню KDE входят разделы или группы (см. рис. 3.14, 3):

- наиболее часто используемые приложения;
- все приложения;
- действия.

Путем использования Редактора меню можно самостоятельно добавлять новые приложения. При движении указателя мыши по меню разворачиваются подпункты. Если найдена программа, которую следует запустить, необходимо щелкнуть по ней мышью.

Перечислим вкратце некоторые из Приложений и Действий:

 **Office** — кнопка вызова программных средств подготовки документов, презентаций, обработки электронных таблиц и пр. В качестве одного из популярных приложений в системе фигурирует OpenOffice.org Writer (см. рис. 3.14, 4, рис. 3.27);

 **SUSE Help Center** — вызов заглавной страницы центра справки KDE (см. рис. 3.16);

 **Поиск файлов и папок** — кнопка запуска KFind. Поиск может осуществляться как по фрагменту имени и расположению (рис. 3.17, поиск файлов с расширением .txt на флэш-накопителе GOGA_1GB), так и по более сложным критериям:

- по содержанию файла и сведениям о документе (вкладка Содержимое);
- по интервалу дат, размеру и пользователям файла (вкладка Свойства);

 **Выполнить программу...** — кнопка «быстрой командной строки» (тот же эффект дает нажатие <Alt+F2>) вызывает маленькое окно (рис. 3.18, а), в котором можно ввести команду Unix, запускающую соответствующую программу (следует учесть, что при этом не будет виден текст, выводимый программой, после ее запуска из командной строки, в отличие от использования терминала — см. рис. 3.6);

 **Завершить сеанс...** — выход из KDE. При нажатии данной кнопки появляется меню (рис. 3.18, б), отражающее возможности — завершить сеанс, выключить, перезагрузить или перевести в ждущий режим компьютер.

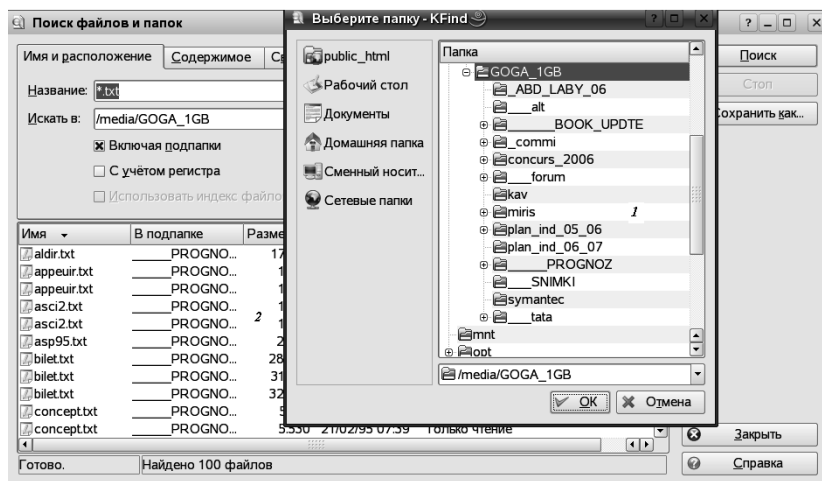
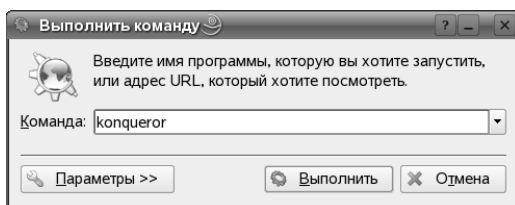
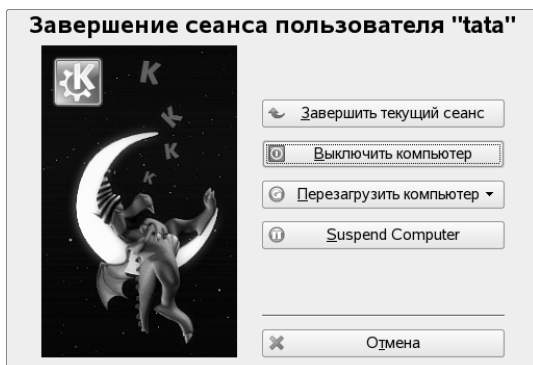


Рис. 3.17. Окно поисковой программы KFind:

1 — выбор места поиска (диск, каталог); 2 — результат поиска файлов *.txt



а



б

Рис. 3.18. Окно командной строки (а); выход из KDE (б)

Работа с окнами

Заголовок окна (рис. 3.19, 1). В заголовке содержится имя окна. При щелчке правой кнопки мыши появляется меню управления окном (рис. 3.19 2, 3). Если программа перестала реагировать, вы можете закрыть ее (что иногда дает возможность сохранить сделанную работу).

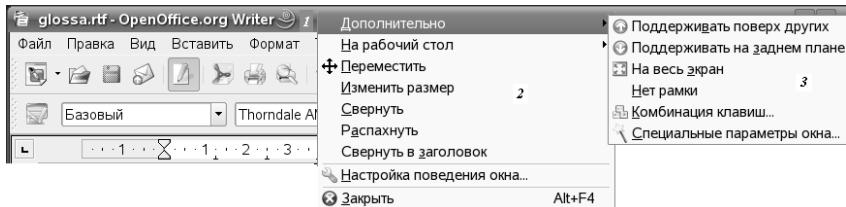


Рис. 3.19. Заголовок и меню управления окном:

1 — заголовок окна; 2 — основная часть меню; 3 — дополнительная часть

Справа от заголовка окна располагаются три кнопки, которые также могут быть использованы для сворачивания (—), максимизации (□) и закрытия (×) окна (это удобнее, чем через меню управления окном). Свернутые окна могут быть восстановлены щелчком по их иконке на панели задач (см. рис. 3.14, 8).

Для того чтобы передвинуть окно, следует поместить курсор на заголовок окна и нажать левую кнопку мыши, а затем, не отпуская кнопку, передвинуть окно и отпустить кнопку. Если вы хотите изменить размер окна, передвиньте курсор на рамку окна. Курсор поменяет свою форму на двойную стрелку (↔). Удерживая нажатой левую кнопку мыши, следует передвинуть рамку в нужном направлении и отпустить кнопку. Если же «ухватиться» мышью за угол окна, то этим размер можно изменять одновременно в двух направлениях.

Меню управления окном. При щелчке правой кнопкой мыши по заголовку окна вызывается меню управления окном (рис. 3.19, 2, 3). Это же меню появляется при нажатии правой кнопки мыши на иконке приложения в панели задач. В нем содержатся команды, с помощью которых можно управлять данным окном (табл. 3.8).

«Горячие клавиши» и использование мыши. В табл. 3.9 приведены комбинации клавиш, которые можно использовать как для управления окнами, так и приложениями, в частности при работе в полноэкранном режиме.

Таблица 3.8. Меню управления окном

Пункт меню	Действие	
Переместить	Щелкнуть по нему левой кнопкой мыши, чтобы передвинуть окно	
Размер	Перемещая указатель мыши, можно изменить его размер, который зафиксирован после щелчка левой кнопкой	
Свернуть	Спрятать окно, оставив только пиктограмму в панели задач (рис. 3.21, 3)	
Распахнуть	Распахнуть окно на весь рабочий стол	
Свернуть в заголовок	«Сворачивает» окно в заголовок, оставляя видимым только сам заголовок (рис. 3.20). Чтобы восстановить окно в прежнем виде, следует выбрать Свернуть в заголовок повторно или дважды щелкнуть по заголовку	
На рабочий стол	Отправляет окно на другой рабочий стол. Окно исчезнет, и вновь вы его увидите, выбрав нужную кнопку рабочего стола в панели KDE. Если указать Все рабочие столы, то окно будет видно на всех рабочих столах	
Настройка поведения окна...	Этот пункт позволяет изменить облик, а иногда и размещение всех окон KDE (см. рис. 3.24). Изменяются детали графических элементов, например кнопок заголовка, вид кнопок и переключателей в диалогах, полос прокрутки и самого заголовка. Все параметры оконного менеджера можно настроить, включая то, как ведут себя окна при перемещении, как они реагируют на различные щелчки мышью и пр.	
Заккрыть	Заккрыть данное окно. Иногда приложение сможет само сохранить работу при закрытии, но в некоторых случаях этого не происходит. Лучше всего закрывать программы, используя их собственные команды, а команду Заккрыть применять лишь как крайнюю меру. Иногда окно закрывается, но приложение все равно продолжает «висеть» в фоновом режиме, поэтому, вернувшись на рабочий стол, надо проверить список запущенных процессов	
Дополнительно	Поверх остальных	Окно будет все время оставаться поверх остальных окон, даже если оно не активно
	Под всеми остальными	Окно будет все время оставаться ниже всех остальных окон
	Полноэкранный режим	Развернуть окно на весь экран, при этом будут скрыты даже элементы управления окном. Для переключения потребуется работать с клавиатурой (табл. 3.9). Вернуться обратно из полноэкранного режима можно, нажав <Alt+F3> и сделать отмену в меню Дополнительно\Полноэкранный режим

Окончание табл. 3.8

Пункт меню	Действие	
	Без рамки	Убрать из окна рамку и элементы управления. Отменить этот режим можно, опять же нажав <Alt+F3> и зайдя в меню Дополнительно\Без рамки
	Сохранить настройки окна	Сохранить текущие параметры окна (положение, размер, должно ли оно выводиться поверх всех окон и т. д.), чтобы при следующем запуске это окно создавалось с указанными параметрами

Таблица 3.9 «Горячие клавиши» KDE

Клавиши	Действие
<Alt+Tab> или <Alt+Shift+Tab>	Переключение между окнами
<Ctrl+Tab> или <Ctrl+Shift+Tab>	Переключение между рабочими столами
<Alt+F2>	Быстрая командная строка
<Alt+F3>	Меню управления окном
<Alt+F4>	Заккрытие окна
<Ctrl+F1..F8>	Переключение на указанный рабочий стол
<Ctrl+Alt+Esc>	Заккрытие выбранного окна (то окно, на котором щелкните мышью, будет закрыто). Нажмите эту комбинацию клавиш еще раз, чтобы восстановить нормальный вид курсора
<Ctrl+Alt+BS>	Выход из KDE (без сохранения!). Используйте только в крайнем случае
<Ctrl+Alt+Num+>	Переключение на следующее разрешение экрана (необходимо настроить X-сервер, чтобы это работало)
<Ctrl+Alt+Num—>	Переключение на предыдущее разрешение экрана (необходимо настроить X-сервер, чтобы это работало)



Рис. 3.20. Сворачивание окна в заголовок



Рис. 3.21. Панель задач:

1 — окно на переднем плане; 2 — на заднем плане; 3 — свернуто в кнопку

Для управления окнами можно использовать также манипуляции мышью (табл. 3.10).

Таблица 3.10. Использование мыши*

Действие	Результат
Щелчок по рамке окна или его заголовку: левой кнопкой мыши	Активирует окно и выводит его на передний план поверх остальных окон
средней кнопкой	Прячет окно на задний план
правой кнопкой	Показывает меню управления окном, если окно активно, если нет — активирует его
Двойной щелчок по заголовку окна	Сворачивает окно («как жалюзи»), так что виден только его заголовок
Перетаскивание заголовка окна	Перемещает окно
Перетаскивание углов или границ	Изменяет размер окна
<Alt+Левая кнопка мыши>	Перемещает окно
<Alt+Средняя кнопка мыши>	Перемещает окно на передний план
<Alt+Правая кнопка мыши>	Растягивает окно в направлении движения мыши
Щелчок по левой верхней пиктограмме	Закрывает программу
Щелчок по кнопке <i>Распахнуть</i> : левой кнопкой мыши	Распахивает окно
средней кнопкой	Распахивает только по вертикали
правой кнопкой	Распахивает по горизонтали

* Это настройки по умолчанию, которые могут быть изменены (см. табл. 3.11).

Настройки интерфейса

Рассмотрим некоторые возможности настройки интерфейса.

Настройка панели KDE. Как упоминалось выше, в Kicker можно добавить множество объектов. На рис. 3.22, 1 показан процесс добавления приложения через меню *Добавить* на панель \Приложение\Office\Word Processor.

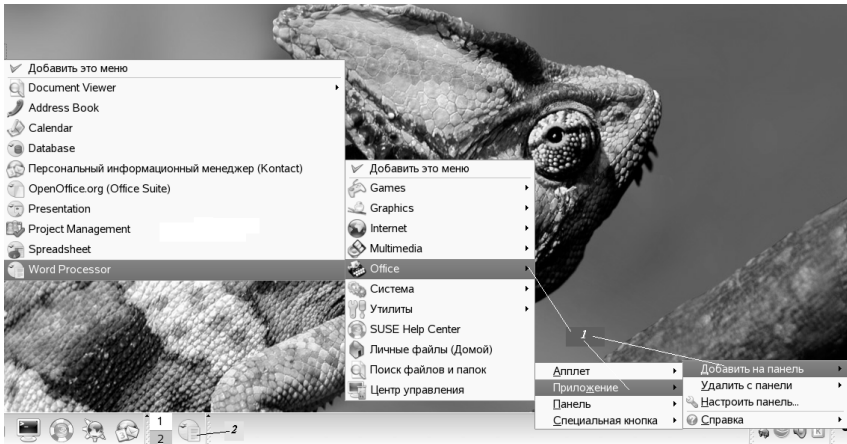


Рис. 3.22. Добавление приложения Word Processor в Панель:
1 — контекстное меню добавления; 2 — пиктограмма добавленного приложения

После выполнения этих действий на панели появляется пиктограмма приложения (рис. 3.22, 2, рис. 3.23, а). В дальнейшем при необходимости приложение может быть удалено с панели (рис. 3.23, б).

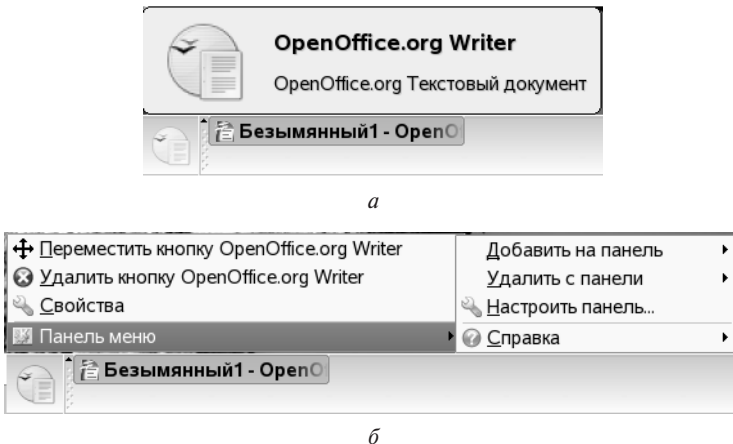


Рис. 3.23. Добавленное приложение с контекстным меню (а); меню удаления приложения с панели (б)

Аналогично на панель могут быть добавлены апплеты, папки и другие объекты.

Настройки окон. KDE предоставляет широкие возможности настройки окон приложений. Вкратце рассмотрим некоторые из них.

Поведение окна. Данный модуль вызывается из меню управлением окном (рис. 3.24), в левой части которого располагаются панели Фокус, Действия, Перемещение и Дополнительно. На панели фокус можно настроить, как окна получают или теряют фокус, т. е. становятся активными или неактивными. «Фокус» имеет окно, с которым в данный момент работает пользователь. Окно с фокусом также называют «активным окном». Фокус — не обязательно окно на переднем плане («поднятое» окно). Фокус и перемещение окон на передний план настраиваются отдельно. Панель Действия устанавливает реакцию окна на щелчки мыши. Панель Перемещение позволяет настроить перемещение окон и их расположение при запуске приложений. Опции панели Дополнительно включают специфические настройки, такие, как перемещение окон между рабочими столами и «shading», т. е. сворачивание окон в заголовок.

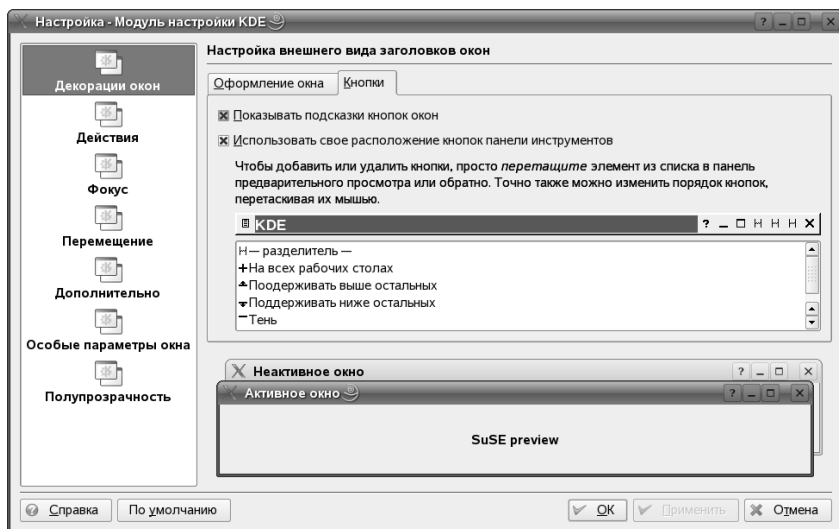


Рис. 3.24. Экран режима Настройка поведения окна

Некоторые из возможных настроек панелей Фокус и Действия приведены в табл. 3.11.

Таблица 3.11. Основные настройки поведения окна

Действие	Комментарий
Установка фокуса	
По щелчку	Окно получает фокус только после щелчка на нем
Фокус следует за мышью	Окно получает фокус при наведении курсора мыши на него. Новые окна, такие как окно командной строки, вызываемое по <code><Alt+F2></code> , получают фокус в любом случае. В других оконных менеджерах эта опция иногда называется «Sloppy focus follows mouse»
Фокус под мышью	Окно, попадающее под курсор мыши, становится активным. Новые окна, например вызванное с помощью <code><Alt+F2></code> окно запуска программы, не получают фокус автоматически
Фокус строго под мышью	Если мышь не находится ни на одном из окон (например, на панели задач), ни одно окно не активно. Новые окна, например вызванное с помощью <code><Alt+F2></code> окно запуска программы, не получают фокус автоматически
Действие: двойной щелчок на заголовке	
Распахнуть	KDE максимизирует окно при двойном щелчке на заголовке. Можно также выбрать увеличение окна только по горизонтали или только по вертикали
Свернуть в заголовок	Окно будет уменьшено до заголовка. Чтобы восстановить окно, нужно будет снова дважды щелкнуть по заголовку
Действие: щелчок по заголовку или рамке	
Активизировать	Сделать окно активным
Убрать вниз	Минимизирует окно. Будет освобожден рабочий стол
Меню операций	Выведет небольшое подменю, в котором вы сможете выбрать команды, связанные с окном (например, Распахнуть, Свернуть, Закрыть и т. д.)
Вывести поверх всех	Поместит окно поверх остальных окон
Активизировать или убрать вниз	Окна, расположенные не на самом верху, будут подняты, расположенные наверху — убраны вниз
Действие: щелчок любой из трех кнопок мыши внутри неактивного окна	
Активизировать, поднять и обработать щелчок	Щелчок активизирует окно, поместит его поверх остальных окон. Также он будет обработан приложением, запущенным в этом окне
Активизировать и обработать щелчок	Щелчок активизирует окно и будет обработан приложением, запущенным в этом окне
Активизировать	При этом щелчок просто активизирует окно. Он не будет обработан приложением

Окончание табл. 3.11

Действие	Комментарий
Активизировать и вывести поверх всех	При щелчке окно будет активизировано и помещено поверх всех окон. Щелчок не будет обработан приложением
Действие: щелчок мыши при нажатом модификаторе (по умолчанию <Alt>) внутри окна, заголовка или рамки	
Убрать вниз	Минимизирует окно. Будет освобожден рабочий стол
Переместить	Позволит перемещать выбранное окно по рабочему столу
Вывести поверх всех	Поместит окно поверх остальных окон
Изменить размер	Вы сможете изменить размер выбранного окна
Вывести поверх всех / убрать вниз	Окна, расположенные не на самом верху, будут подняты, расположенные наверху — убраны вниз

Панель инструментов. Щелчком по правой кнопке мыши на панели инструментов вызывается Меню панели инструментов (рис. 3.25), которое позволяет:

- изменять размещение панели;
- изменять размер значков;
- снабжать значки подписями и пр.

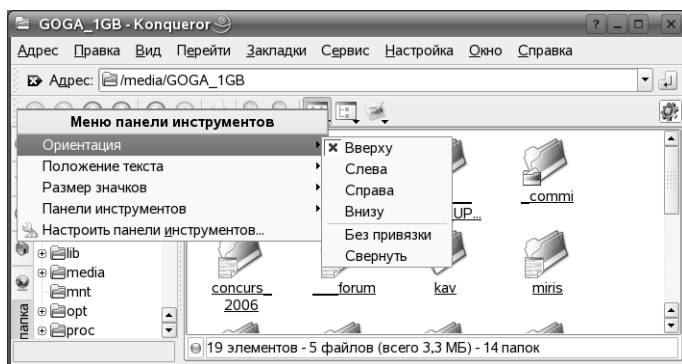
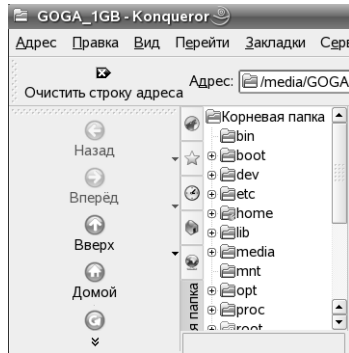


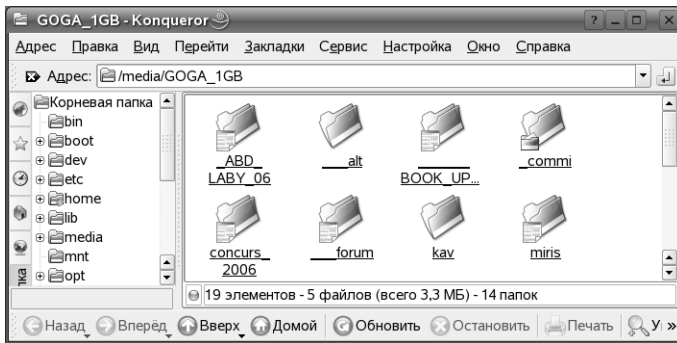
Рис. 3.25. Меню панели инструментов

На рис. 3.26 приведены примеры различных размещений панели инструментов в окне приложения:

- а* — панель слева, подписи под значками;
- б* — панель снизу, подписи рядом со значками.



а



б

Рис. 3.26. Примеры управления размещением панели инструментов:

а — слева; б — снизу

Некоторые приложения KDE



Редактор документов OpenOffice.org Writer.

Работая в OpenOffice.org Writer, пользователь может создавать любые текстовые документы, составлять личные и официальные письма, брошюры, факсы и профессиональные учебные пособия. Документы, которые используются часто, можно сохранять как шаблоны, например бланк счета. Имеется проверка орфографии и тезаурус, а при необходимости может быть задействована Автозамена и расстановка переносов во время ввода текста с клавиатуры.

В OpenOffice.org нет ограничений на длину текстового документа (рис. 3.27).

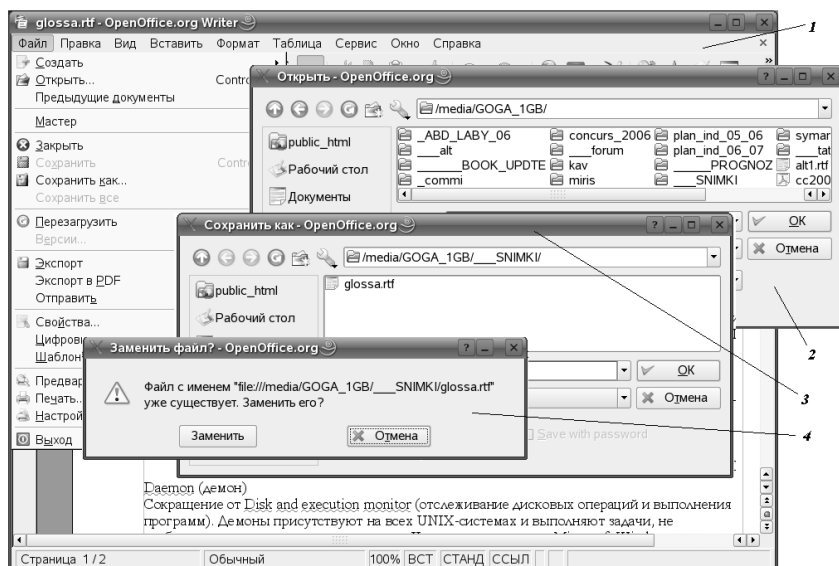


Рис. 3.27. Экраны редактора OpenOffice.org Writer:

1 — окно приложения; 2 — окно открытия файлов; 3 — окно сохранения файлов; 4 — запись новой редакции файла

Создание и структурирование. OpenOffice.org предлагает большой выбор средств для создания документов. Используя Мастер стилей, можно создавать и редактировать стили абзацев, отдельных символов, рамок и страниц. Навигатор поможет при создании структурированных текстов, а также позволит отредактировать уже созданную структуру текста путем перемещения абзацев из одного места в другое.

Можно создавать разнообразные указатели и таблицы в текстовых документах, а структуру и внешний вид этих указателей и таблиц задавать в соответствии с потребностями. Активные гиперссылки и закладки позволяют переходить к соответствующим местам в тексте.

Подготовка публикаций с помощью OpenOffice.org Writer. OpenOffice.org Writer содержит многочисленные возможности для создания профессиональных документов. Текст может быть многостолбцовым и содержать текстовые рамки, рисунки,

таблицы и пр., встроенные в него. Текстовые рамки могут создавать газетный формат. Такие возможности, как линии выравнивания, рамки для иллюстраций выборочно и по всему документу и выбор любого цвета для символов, абзацев и таблиц, помогают создавать различные документы на высоком профессиональном уровне.

Вычисления. Текстовые документы в OpenOffice.org имеют встроенную функцию вычисления, которая позволяет выполнить математические действия или создать логические связи. Чтобы выполнить вычисления, в текстовом документе можно создать таблицу.

Создание чертежей. Удобный и легко доступный инструмент для черчения позволяет создавать чертежи, рисунки, врезки и пр. непосредственно в текстовых документах.

Вставка изображений. Предусмотрена возможность вставки в текстовый документ картинок различного формата, включая jpg, gif и пр. Наиболее распространенные форматы могут быть отредактированы непосредственно в текстовом документе с помощью графического редактора. Кроме этого, Галерея предоставляет коллекцию тематически упорядоченных картинок.

Изменяемый интерфейс приложения. Интерфейс программы разработан таким образом, что каждый пользователь может настроить его в соответствии с личными предпочтениями. Различные окна (Мастер стилей, Навигатор и т. п.) могут быть размещены как плавающие окна в любом месте экрана, а некоторые из них могут быть пристыкованы к краю. Значки и меню также могут быть изменены.

Перетаскивание. Возможность перетаскивания позволяет работать с текстовым документом в OpenOffice.org более быстро и эффективно. Например, можно перетаскивать картинки непосредственно из Галереи в активный документ.

Полнообъемная справка. Программа предоставляет Справку, которая служит гидом по элементам программы в OpenOffice.org, и включает в себя большой набор инструкций для решения простых и сложных задач.

Графический редактор OpenOffice.org Draw.

Работа с векторной графикой. OpenOffice.org Draw является объектно-ориентированной программой для создания векторной графики. Объектами могут быть линии, прямоугольники, трехмерные цилиндры или любые многоугольники. Для

всех объектов уже установлены параметры, такие как размер, цвет поверхностей и контуров, привязанные файлы и т. д. Все параметры могут быть изменены (рис. 3.28).

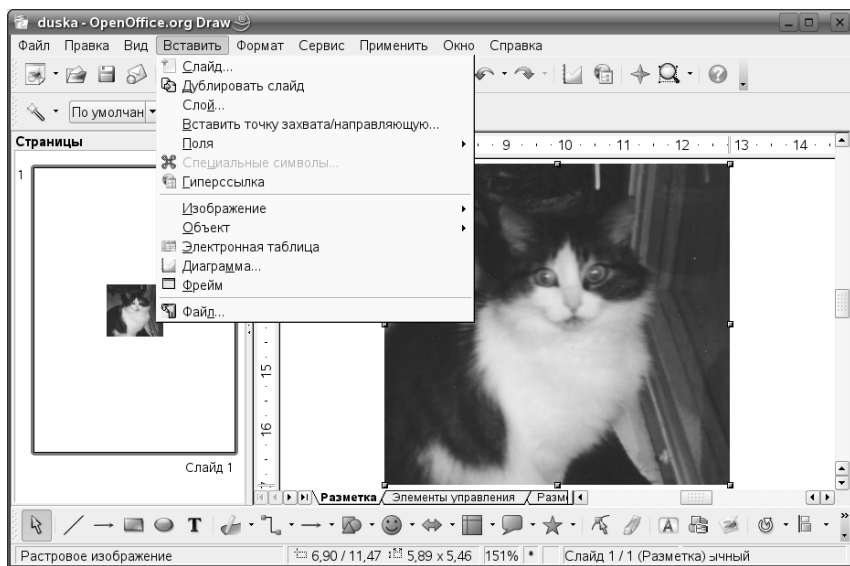


Рис. 3.28. Экран приложения OpenOffice.org Draw

Благодаря векторной технологии, можно вращать объекты в любом направлении и изменять их размер. Объекты можно двигать и частично совмещать, так как все они контролируются индивидуально.

Работа с трехмерными объектами. Пользователь не ограничен двумерностью при использовании OpenOffice.org Draw. Можно создавать кубы, сферы, цилиндры и другие трехмерные объекты, вращать их и даже подсвечивать с помощью различных эффектов. Используя эти параметры, можно быстро создавать презентации.

Работа с диаграммами. OpenOffice.org Draw объектно-ориентирован. Это дает возможность создавать прямоугольники, содержащие текст и связанные друг с другом. При перемещении фигур связи автоматически сохраняются, что упрощает рисование и работу с презентациями и диаграммами. Связи размещаются между точками соединения, что заметно облегчает создание технических рисунков с текстовыми пояснениями.

Рисование. При обычном рисовании используются прямые, простые линии, кривые Безье и различные виды прямоугольников и прочих геометрических фигур. Трехмерные параметры могут быть использованы для создания трехмерных объектов, например кубов, сфер и торов. Также возможно вращение двумерных форм в трехмерном измерении. Выбирая цвет с расстановкой структурой, перспективой и освещением, можно создать трехмерные объекты для приглашений, брошюр и визитных карточек.

Для вставки в рисунки дополнительных элементов используйте коллекцию картинок, находящихся в галерее. Не имеет значения, состоят ли графические объекты из векторов или точек.

Экспорт. OpenOffice.org Draw может использоваться для создания кнопок и значков для Web-страниц и экспорта их в форматах gif, jpg, png и других.

Создать. Программа предлагает ряд подсказок для создания собственных рисунков. Можно определить сетку, к которой объект будет привязан во время создания или перемещения либо временно привязать несколько новых объектов к границам и узлам уже существующих объектов. Размер объекта можно изменить в любое время с помощью ввода новых размеров в окне диалога Параметры.

Интегрировать. Предусмотрена возможность импорта текстов, таблиц, диаграмм, формул из других программных модулей OpenOffice.org в рисунок.

Редактор электронных таблиц OpenOffice.org Calc позволяет вычислять, анализировать и преобразовывать данные в электронной таблице. Могут быть импортированы и обработаны также таблицы Microsoft Excel (рис. 3.29).

Вычисления. OpenOffice.org Calc позволяет использовать функции (включающие статистические, финансовые, банковские операции), которые можно использовать для построения сложных формул обработки данных. Для построения формул предлагается встроенный Мастер функций.

Динамические вычисления. Система предоставляет возможность немедленно увидеть, каким образом меняются результаты вычислений, подверженные влиянию множества факторов, при условии изменения одного из этих факторов. Предусмотрена возможность построения обширных таблиц, отражающих изменения переменных в различных сценариях.

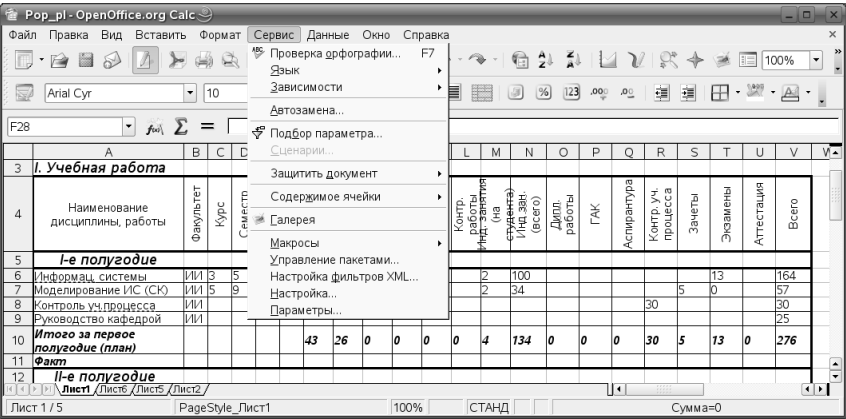


Рис. 3.29. Экран приложения OpenOffice.org Calc

Функции базы данных. Предусмотрены возможности хранения, фильтрации и сортировки данных в таблицах, а также импорт содержания таблиц из внешних баз данных.

Организация данных. При работе с таблицами можно легко скрыть или показать данные, отвечающие определенным критериям, а также форматировать данные и определять итоговые суммы и подсуммы при сортировке строк.

Динамические диаграммы. OpenOffice.org Calc позволяет представлять данные из таблиц в форме разнотипных диаграмм, которые автоматически обновляются при изменении данных.

Открытие и сохранение файлов Microsoft. OpenOffice.org Calc позволяет конвертировать файлы Excel (см. рис. 3.29), а также открывать и сохранять их во множестве различных иных форматов.

Список наиболее часто используемых комбинаций клавиш в OpenOffice.org приводится в табл. 3.12.

Таблица 3.12. Использование клавиш в OpenOffice.org

Комбинация клавиш	Эффект
<Enter>	Нажимает кнопку диалога с фокусом
<Esc>	Закрывает действие или диалог. В OpenOffice.org Help — на один уровень вверх. Курсор расположен в поле URL панели функций, курсор возвращается в документ. Если URL был выделен, следует нажать эту клавишу дважды

Продолжение табл. 3.12

Комбинация клавиш	Эффект
<Space> Пробел	Переключает флажок с фокусом в диалоге
Клавиши управления курсором	Изменяют активное поле в поле выбора диалога
<Tab>	Перемещает фокус к следующему разделу или элементу диалога
<Shift+Tab>	Перемещает фокус к предыдущему разделу или элементу диалога
<Alt+↓>	Открывает список текущего выделенного поля управления. Эти комбинации клавиш применимы не только к полям со списком, но и к кнопкам со значками со всплывающими меню. Закройте открытый список, нажав клавишу <Esc>
<Del>	Удаляет выделенный элемент(ы) в мусорную корзину
<Shift+Del>	Удаляет выделенный элемент(ы) без помещения их в мусорную корзину
<Backspace>, <BS>	Если открыта какая-либо папка, возвращает на один уровень вверх (назад)
<Ctrl+Tab>	Переход к следующему открытому документу (за исключением случая, когда курсор расположен в начале заголовка, вместо этого вставляется символ табуляции)
<Shift+Ctrl+Tab>	Переход к предыдущему открытому документу
<Ctrl+O>	Открывает документ
<Ctrl+S>	Сохраняет текущий документ
<Ctrl+N>	Создает новый документ
<Shift+Ctrl+N>	Открывает диалог Шаблоны и документы
<Ctrl+P>	Печатает документ
<Ctrl+Q>	Выход из приложения
<Ctrl+X>	Вырезает выделенные элементы
<Ctrl+C>	Копирует выделенные элементы
<Ctrl+V>	Вставка из буфера обмена
<Ctrl+A>	Выделяет все
<Ctrl+Z>	Отменяет последнее действие
<Ctrl+F>	Вызывает диалог Найти и заменить
<Ctrl+Shift+F>	Поиск последнего введенного искомого понятия

Окончание табл. 3.12

Комбинация клавиш	Эффект
<Ctrl+Shift+J>	Переключает вид между полноэкранным режимом и обычным
<Ctrl+Shift+R>	Повторно открывает окно документа
<Ctrl+I>	К выделенной области применяется атрибут Курсив. Если курсор расположен внутри слова, это слово также выделяется курсивом
<Ctrl+B>	К выделенной области применяется атрибут Полужирный. Если курсор расположен внутри слова, оно также становится Полужирным
<Ctrl+U>	К выделенной области применяется атрибут Подчеркнутый. Если курсор расположен внутри слова, это слово также становится подчеркнутым
<Ctrl+Shift+O>	Устанавливает курсор в поле Загрузить URL на панели функций
<Alt+O>	В диалоге проверки орфографии слово, отмеченное первоначально, как неизвестное/неправильное, будет перенесено во входную строку (слово)
<F1>	Запускает Справку OpenOffice.org
<Shift+F1>	Контекстная справка
<Ctrl+F4> или <Alt+F4>	Закрывает текущий документ (закрывает OpenOffice.org, если последний открытый документ к этому времени уже закрыт)
<F6>	Перемещение фокуса в следующее окно (например, документ / вид источника данных)
<Shift+F6>	Установка фокуса на предыдущее окно
<Shift+F10>	Открывает контекстное меню



Графический редактор GIMP (GNU Image Manipulation Program). Как следует из названия, это программа для манипуляций изображениями. По возможностям GIMP схож с редакторами PaintShop Pro и Adobe PhotoShop.

При первом запуске производится начальная настройка программы. Далее появляется основная панель инструментов, в верхней части которой расположено меню (рис. 3.30).

Выбрав в меню Файл команды Новый или Открыть, можно создать новое изображение или загрузить его с диска.

В окне редактирования можно вызвать меню с помощью правой кнопки мыши (рис. 3.31).

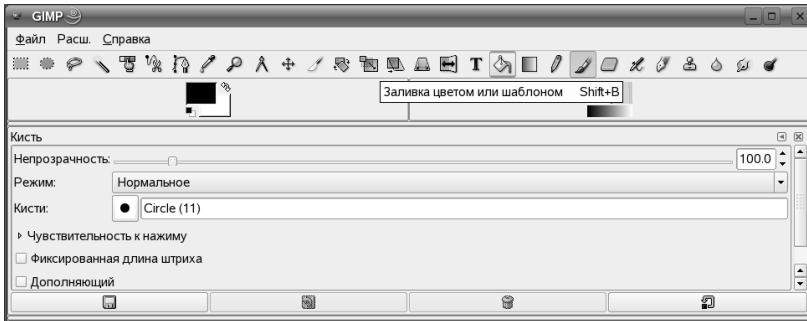


Рис. 3.30. Панель инструментов GIMP

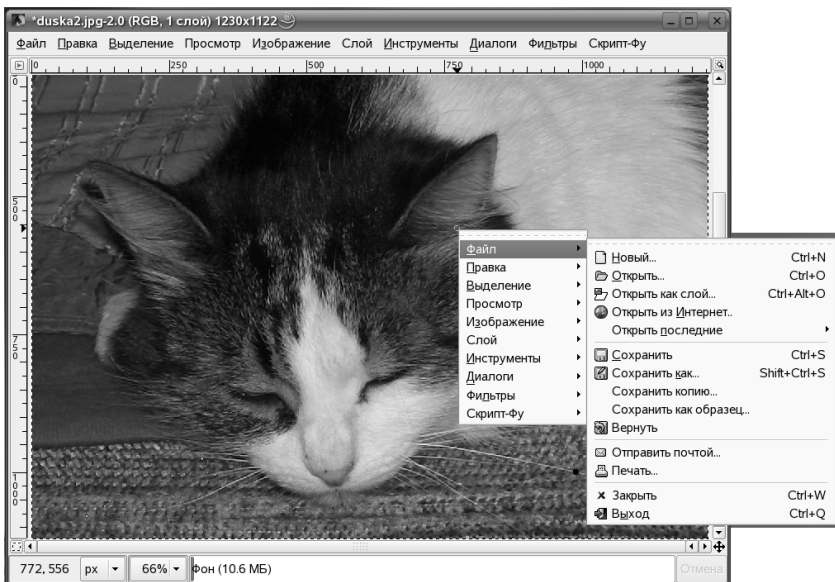


Рис. 3.31. Экран редактора GIMP

Форматы PostScript и PDF. Формат PostScript был разработан в середине 80-х годов фирмой Adobe как язык управления принтерами. Поскольку в то время Unix использовался исключительно на мощных и дорогих компьютерах, то к ним обычно были подключены дорогие PostScript-принтеры. Поэтому термин «печать» в графических программах под Unix обычно означает выдачу данных в формате PostScript, которые потом либо отправля-

ются на принтер (если он не поддерживает PostScript, то фильтр принтера автоматически преобразует их в нужный формат), либо сохраняются в файле с расширением .ps.

Достоинством PostScript является его универсальность — файл, созданный в одной системе, можно использовать и в другой. К недостаткам относится, во-первых, большой объем (особенно если в файлах есть растровые изображения), а во-вторых, то, что программы просмотра показывают файл «как картинку» — смотреть можно, а, например, найти определенное слово в тексте нельзя.

Формат PDF (Portable Document Format) был разработан в начале 1990-х той же фирмой Adobe, и, в частности, в нем исправлены перечисленные недостатки PostScript.

Стандартный просмотрщик PostScript — ghostview (KGhostView). GhostView — одна из первых в Unix программа, созданная для просмотра файлов Adobe PostScript. PostScript — основной язык описания страниц для печати в системах Unix, и это приложение может использоваться для чтения документов или для просмотра материалов, предназначенных для печати.

KGhostView отображает и распечатывает файлы PostScript (.ps, .eps) и Portable Document Format (.pdf). Это — перенесенная на KDE программа GhostScript.

Если документ не соответствует стандарту структуры документа Adobe, функциональность программы ограничена. Например, если отсутствует оглавление, становится невозможным отмечать страницы и перемещаться по документу, пропуская их.

Рассмотрим содержимое основного окна KGhostView, действия с помощью кнопок панели инструментов и методы использования списка страниц.

Могут быть открыты несколько окон KGhostView, чтобы просматривать несколько документов. Заголовок окна отображает название документа, открытого в этом окне (рис. 3.32, 1).

В основной части окна отображается страница текущего документа (рис. 3.32, 2). Если страница слишком велика и не помещается в окне, у его границ автоматически размещаются панели прокрутки. Это свойство может быть отключено с помощью меню Настройка.

Панель инструментов и список страниц могут быть скрыты с помощью опций меню Настройка, чтобы освободить пространство экрана для отображения содержимого.

Предусмотрена возможность прокручивать страницу, используя клавиши $\langle \uparrow \rangle$ и $\langle \downarrow \rangle$, или меню Вид\Прокрутить вверх ($\langle \text{Shift} + \text{Space} \rangle$) и Вид\Прокрутить вниз ($\langle \text{Ctrl} + \text{Space} \rangle$), чтобы перемещаться по документу.

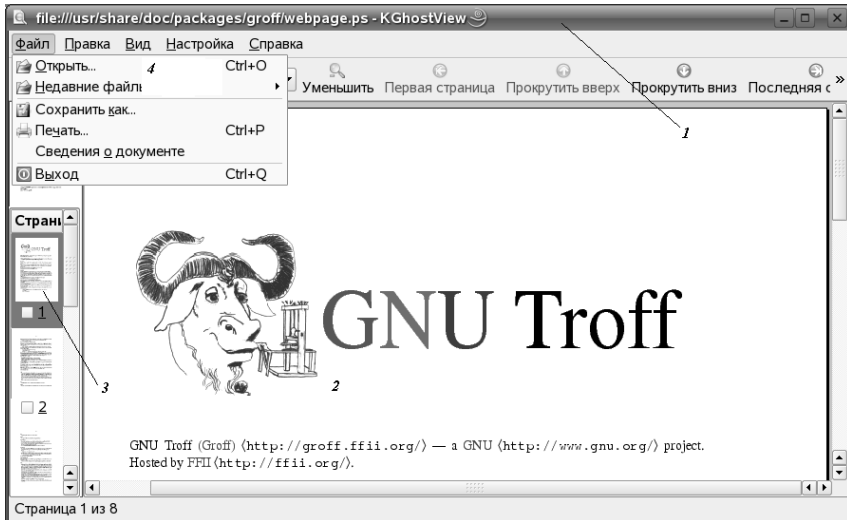


Рис. 3.32. Окно просмотрщика файлов KGhostView:

1 — заголовок; 2 — окно документа; 3 — окно страниц; 4 — меню файл

Кнопки панели инструментов Следующая страница и Предыдущая страница служат для перехода между страницами (также можно щелкнуть левой кнопкой мыши на номере страницы в списке, чтобы перейти на нее).

Список страниц (если выбран показ в меню Настройка) отображается у левой стороны окна (рис. 3.32, 3). Он включает два столбца: первый может содержать флажок отметки страницы, второй отображает номера страниц. Этот список может использоваться, чтобы перемещаться по документу или отмечать страницы для печати.

Может быть установлен «отмечающий страницу» флажок для отображаемой страницы с помощью комбинации клавиш $\langle \text{Ctrl} + \text{M} \rangle$ или щелчком на странице в списке. Вы также можете удалить или изменить пометки, используя меню, выпадающее по щелчку правой кнопки мыши в списке страниц, или с помощью меню Отметки страниц.



Mozilla Firefox. Mozilla Firefox является Internet-браузером, предназначенным для просмотра Web-страниц и поиска в Internet. На рис. 3.33 приводится экран приложения с иллюстрацией некоторых возможностей:

- переход к страницам по закладкам (рис. 3.33, 2) или с использованием журнала посещений (рис. 3.33, 3);
- контекстный поиск по текущей Web-странице (рис. 3.33, 4);
- режим добавления закладок с целью последующего возврата к отмеченным страницам (рис. 3.33, 5).

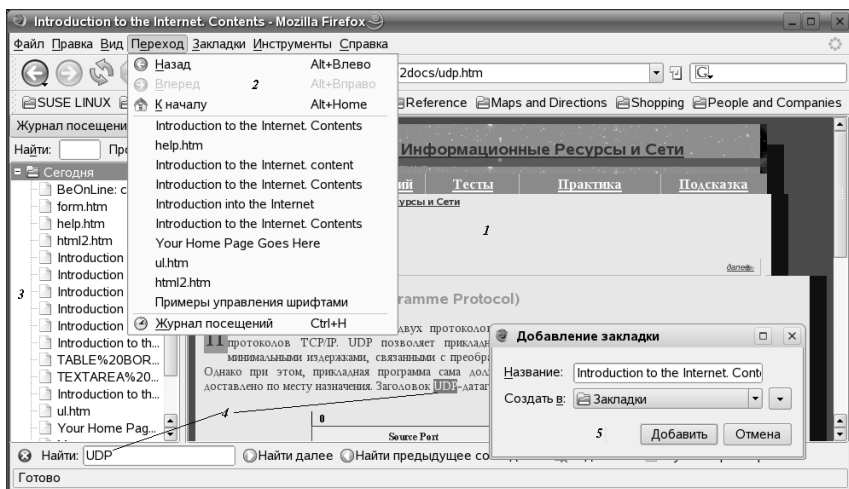


Рис. 3.33. Экран браузера Mozilla Firefox:

1 — окно текущей Web-страницы; 2 — пункт меню Переход; 3 — Журнал посещений (сегодняшний день); 4 — контекстный поиск; 5 — режим добавления закладок

Средства помощи

Центр справки KDE. Доступ к центру справки KDE может быть получен следующим образом (табл. 3.13).

Подсказки и команда «Что это?» Кроме перечисленных выше, предусмотрены и другие пути получения помощи. Если включен режим отображения подсказок (Главное меню\Центр управления\KDE\Внешний вид\Стиль), то каждый раз при размещении указателя мыши на какой-либо кнопке панели инструментов или навигации можно увидеть текст с кратким описанием ее функций (рис. 3.35, а).

Таблица 3.13. Вызов центра справки

Доступ	Примечания
Из меню Справка	Распространенный способ — для того чтобы открыть файл помощи для того или иного приложения, необходимо выбрать в меню приложения пункт Справка (рис. 3.34)
Из главного меню	Если выбрать пункт SUSE Help Center, появится главная страница системы помощи KHelp Center (рис. 3.15)
На панели	Панель Kicker по умолчанию содержит пиктограмму, вызывающую KHelp Center
Из командной строки	Можно запустить KHelp Center с помощью ссылки URL. Ссылки URL добавлены также и для страниц info и man

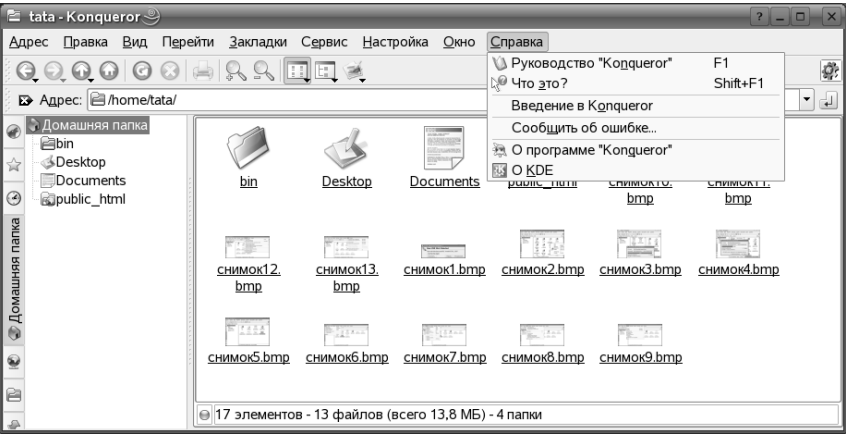


Рис. 3.34. Меню Справка

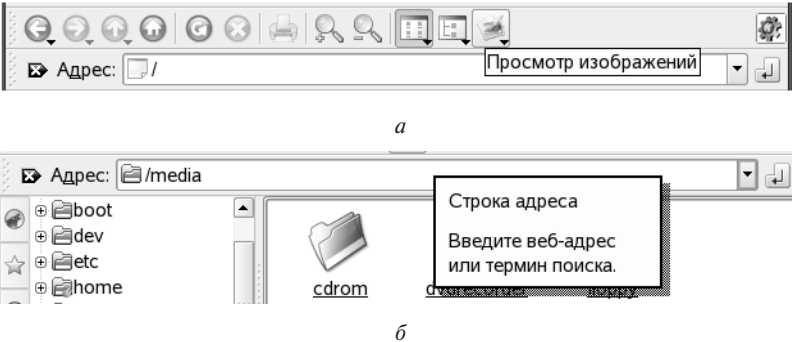


Рис. 3.35. Краткое описание функций (а); справка об элементе Строка адреса (б)

Чтобы получить справку о каком-либо элементе окна, можно также либо выбрать пункт меню *Помощь\Что это?*, либо нажать комбинацию клавиш `<Shift+F1>`, либо кнопку со знаком вопроса в верхнем правом углу окна. Указатель мыши примет форму стрелки со знаком вопроса. После этого надо щелкнуть по тому элементу, по которому следует получить справку, и будет представлено достаточно подробное описание его функций (рис. 3.35, б).

Просмотр файлов справки и руководств (Man и Info). С помощью Konqueror можно просматривать файлы справки по KDE и руководства по Unix (Man и Info), не запуская дополнительных приложений. Чтобы получить справку по какому-либо приложению KDE, необходимо ввести в строку адреса `help:/название программы` (например `help:/ls`).

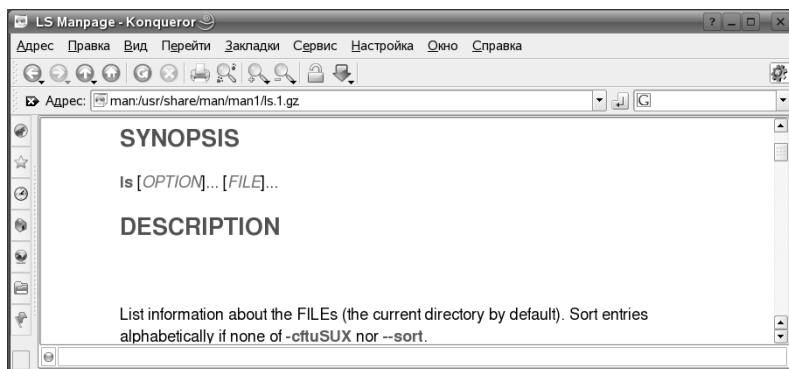


Рис. 3.36. Просмотр руководства Unix (Man); команда `ls`

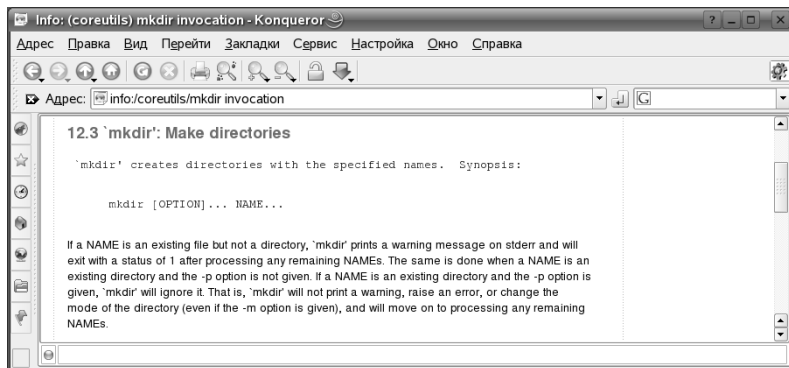


Рис. 3.37. Просмотр документации Unix (Info); команда `mkdir`

Konqueror также позволяет просматривать руководства Unix (Man), например, чтобы просмотреть руководство по команде `ls`, достаточно ввести в строку адреса `man:ls` или `#ls` (рис. 3.36).

Чтобы просмотреть документацию Unix (Info), необходимо ввести в строке адреса `info:/dir`, и Konqueror покажет список всех доступных файлов документации (рис. 3.37).

Необходимо отметить, что файлы справки KDE хранятся в формате, который невозможно просматривать с помощью других обозревателей, кроме Konqueror.

KDE 4

Версия KDE 4 поставляется совместно с ОС SuSE Linux 11.0 и представляет собой дальнейшее развитие продукта. На рис. 3.38 представлены некоторые из элементов рабочего стола (все списки могут быть изменены).

Пиктограммы приложений и служб в данном примере включают следующий перечень (рис. 3.38, 3.39):

 Интерактивная помощь — связь с сайтом help.opensuse.org;

 Мой компьютер — справка о комплектации и программных средствах (рис. 3.38, 4);

 Офис — запуск компонент OpenOffice.org — Writer, Calc, Draw, Мастер презентаций, Мастер базы данных, Writer/Web (создание HTML-документов), Math (редактор формул) и др.;

 Web-обозреватель Firefox;

 Корзина — стандартное временноеместилище удаленных объектов;

 OpenSuse — справка о проекте.

Главное меню содержит закладки (рис. 3.39):

 Избранное может включать, например, обозреватель Firefox, текстовый редактор, аудиоплеер, обработчик фотографий, файловый менеджер (Dolphin), вызов справки;

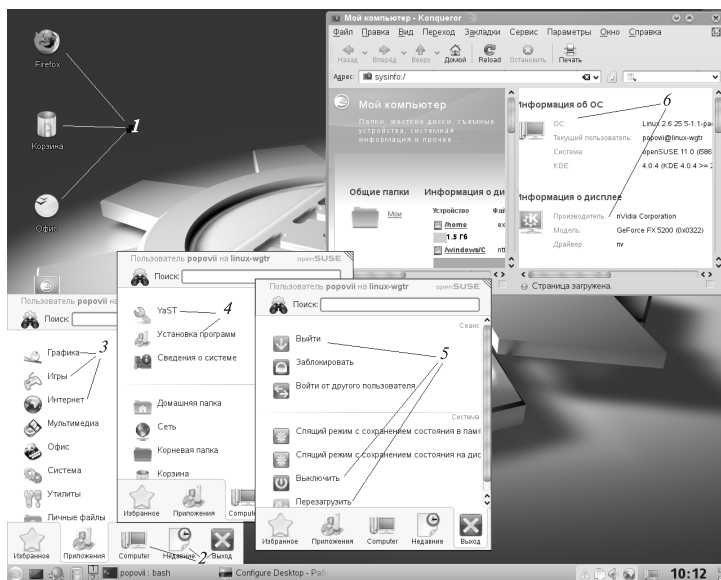


Рис. 3.38. Элементы рабочего стола KDE 4:

1 — пиктограммы приложений и служб; 2 — главное меню; 3 — подменю/закладка Приложения; 4 — закладка Computer; 5 — закладка Выход; 6 — окно Мой компьютер

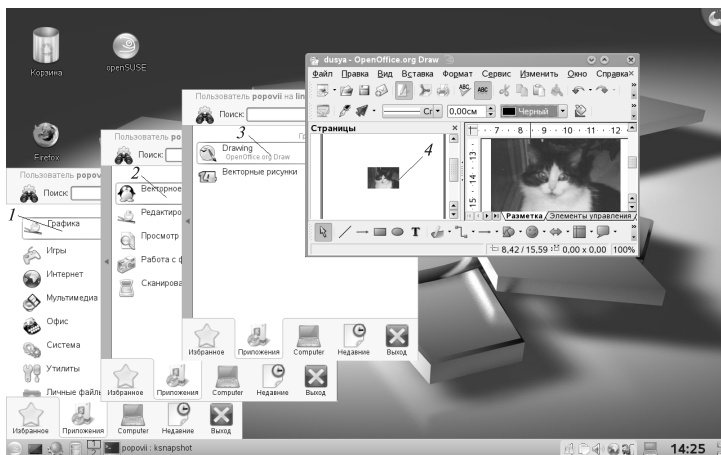


Рис. 3.39. Запуск приложения в KDE 4:

1 — закладка Приложения главного меню, выбор рубрики Графика; 2 — выбор рубрики Векторное рисование; 3 — выбор редактора Drawing; 4 — экран приложения Drawing



— например, группы/рубрики Графика, Игры, Интернет, Мультимедиа, Офис, Система, Утилиты, Личные файлы, Поиск файлов и папок, Системные параметры. **Рисунок 3.39** иллюстрирует запуск одного из приложений — Приложения\Графика\Векторное рисование\Drawing;



— Параметры администратора, Установка программ, Сведения о системе, некоторые узлы файловой системы — Домашняя папка (в данном примере /home/popovii), Сеть (remote/), Корневая папка (/), Корзина (trash/);



— список и пиктограммы ранее запускавшихся приложений и сервисов/утилит;



— перечень различных вариантов завершения работы (Выключить, Перезагрузить, Спящий режим, Выйти, Войти от другого пользователя и пр.).

Взаимодействие Windows и Linux

Рассмотрим ситуацию, когда операционные системы Linux и Windows 2000 установлены на одном компьютере. На рис. 3.40, *а* представлена схема использования дискового пространства до установки, а на рис. 3.40, *б* — после установки Linux. Здесь раздел в 392 Мбайта предназначен для своп-области, а раздел в 18,46 Гбайта — для размещения файловой системы ОС (рис. 3.40, *б*, 2). Остальное — разделы Windows *c:* и *d:* (рис. 3.40, *б*, 1).

Заметим, что ОС Windows 2000 «не видит» подробностей Linux-раздела. На рис. 3.40, *в* представлена вкладка Управление компьютером\Управление дисками\Свойства, на которой для основного раздела Linux сообщается — Файловая система: неизвестно; Занято: 0 байт; Свободно: 0 байт; Емкость: 0 байт.

Между тем при запуске Linux может быть предусмотрено автоматическое монтирование Windows-дисков (в нашем примере *c:* и *d:* подключены в качестве подкаталогов каталога windows — рис. 3.41, 1). Вследствие этого возможно обращение и навигация по диску (рис. 3.41, 2), открытие и корректировка

(редактором OpenOffice.org Writer) файлов, созданных в MS Word (рис. 3.41, 3).

На рис. 3.42 приведена некоторая последовательность команд, выданных на терминал. Читателю предлагается вернуться к командам Unix и самостоятельно разобраться в данном протоколе, а также выяснить связь между рис. 3.42 и рис. 3.41.

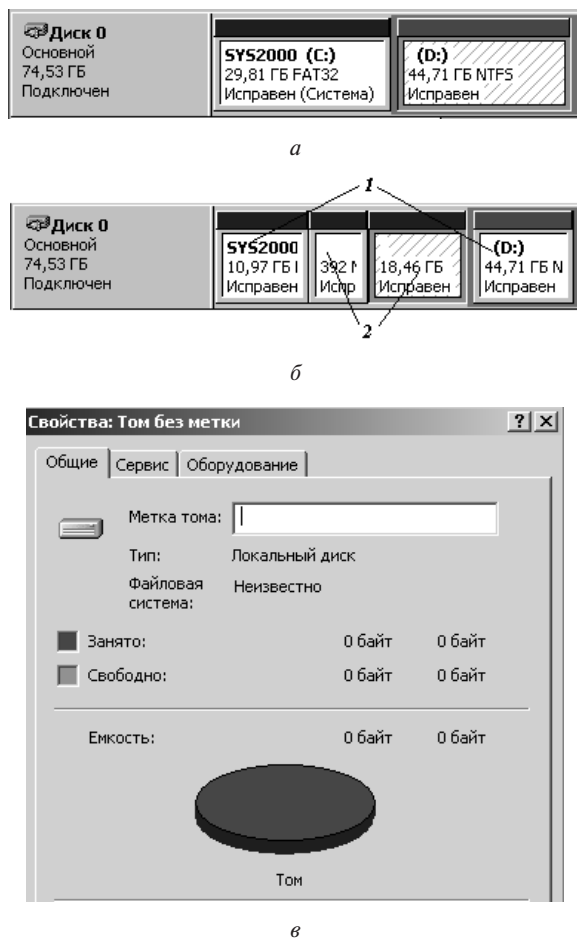


Рис. 3.40. Распределение памяти на жестком диске до установки Linux (а); после установки (б):

1— разделы Windows; 2 — разделы Linux; свойства Linux-раздела с точки зрения службы Управление дисками Windows 2000 (в)

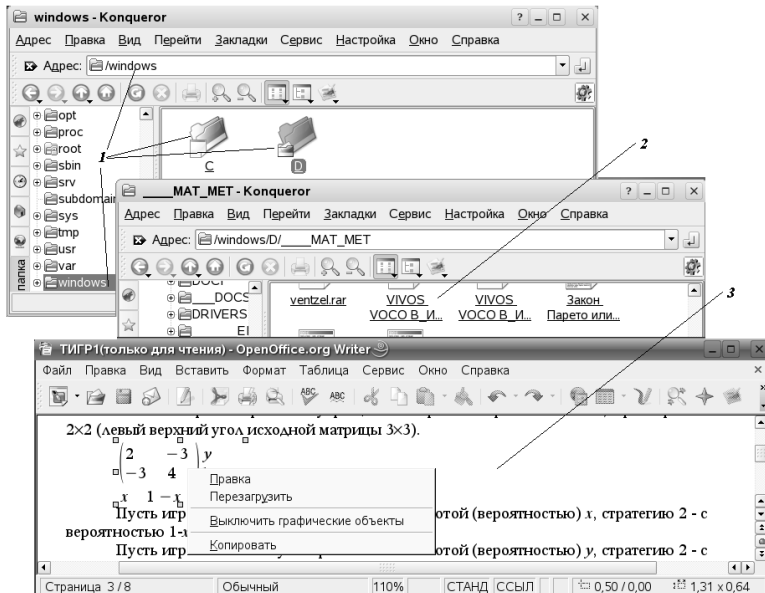


Рис. 3.41. Разделы Windows c: и d:, смонтированные в Linux-каталоге windows (/); просмотр диска d: навигатором Konqueror (2); открытие документа MS Word редактором OpenOffice.org Writer (3)

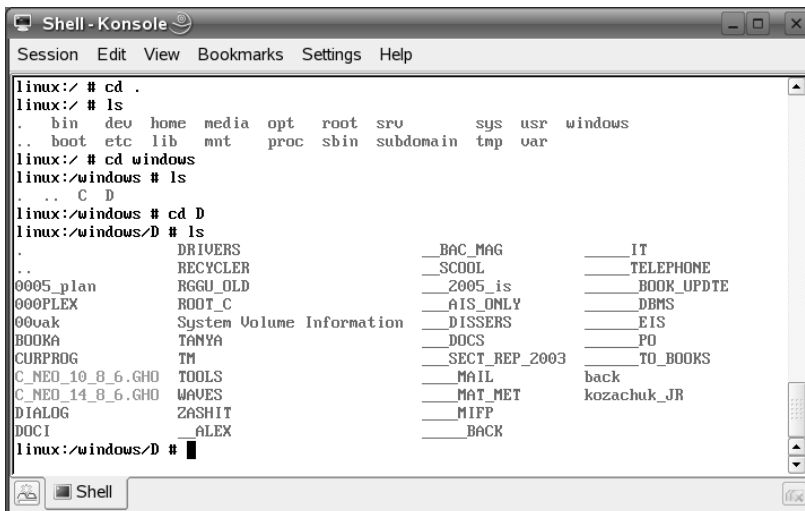


Рис. 3.42. Фрагмент протокола диалога на экране терминала

GNOME (GNU Network Object Model Enviroment)



GNOME — рабочий стол и набор графических инструментальных средств, на основе которых могут создаваться различные приложения. В число основных компонентов рабочего стола входят панель запуска и функциональные средства рабочего стола.

Среда GNOME поддерживает интерфейсы компонентной модели, что позволяет обеспечить взаимодействие программных компонент независимо от языка программирования, на котором они реализованы, или от аппаратной платформы.

Рабочий стол (рис. 3.43) предоставляет пользователю все возможности, присущие современным ОС с графическим интерфейсом — манипулирование файлами, приложениями и каталогами, работа с ними с помощью других приложений и пр.

Допускается перетаскивать файлы из приложения на рабочий стол и наоборот. Доступ к какому-либо приложению или файлу происходит путем перемещения по файлам и каталогам или созданием ссылки. Чтобы создать ссылку, необходимо перетящить пиктограмму файла при нажатой средней кнопке мыши. После этого появится контекстное меню, в котором надо выбрать пункт *Link*. Для создания кнопки запуска приложения нужно щелкнуть правой кнопкой на рабочем столе и выбрать команду *Create Launcher Applet*, после этого откроется диалоговое окно, в котором нужно ввести имя программы, выбрать для нее пиктограмму и права доступа к ней.

В среде GNOME область рабочего стола увеличивается за счет виртуальных столов. Виртуальные столы можно группировать (например, по 4). Каждый виртуальный стол представляет собой отдельный рабочий стол. Данный переключатель находится в нижней части экрана, справа. Количество виртуальных столов может меняться по желанию пользователя. Каждый элемент панели можно перемещать стандартным образом, а также удалять.

Основные элементы рабочего стола GNOME — символические ссылки (ярлыки, иконки) на рабочем столе (рис. 3.43, 1), панели на верхней (рис. 3.43, 2) и нижней (рис. 3.43, 3) границах экрана и меню панелей. Основным инструментом диалога является мышь, хотя GNOME и реализует вспомогательные технологии, например шрифт Брайля, синтезаторы речи и экранную клавиатуру для поддержки пользователей с ограниченными возможностями.

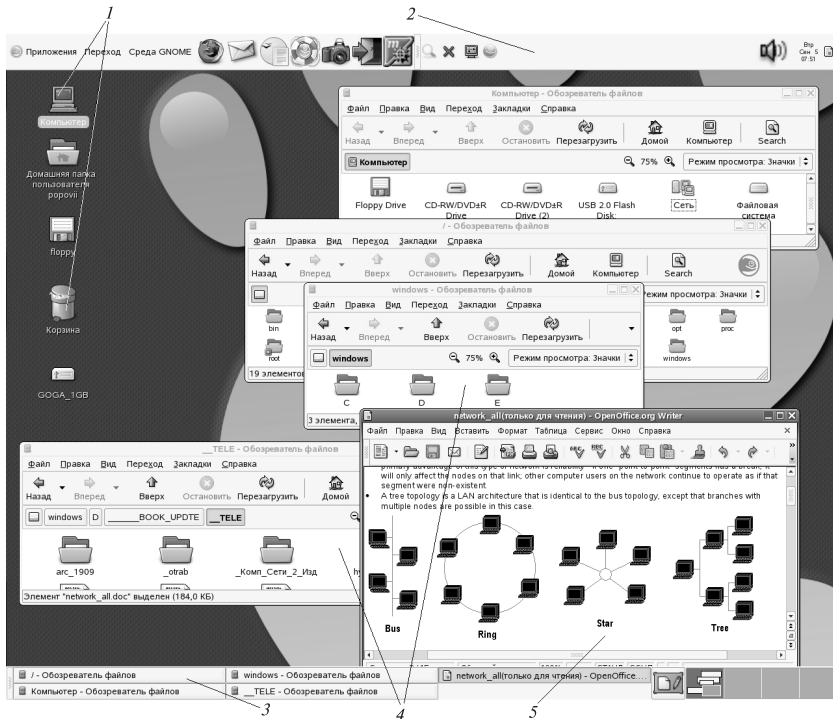


Рис. 3.43. Рабочий стол GNOME:

1 — поверхность стола с ярлыками; 2 — верхняя панель; 3 — нижняя панель; 4, 5 — окна приложений

Некоторые ярлыки рабочего стола. Рабочий стол GNOME показывает следующие изображения (рис. 3.43, 1), обеспечивающие основную навигацию и функциональные возможности системы:



Компьютер — может использоваться для быстрого доступа к любым устройствам, расположенным на компьютере (разделы накопителей на жестких дисках, цифровые камеры и флэш-накопители USB);



Домашний каталог — при входе в систему какого-либо пользователя он получает возможность доступа к каталогу, имя которого совпадает с именем пользователя;



Корзина (мусорное ведро). Для удаления объекта с рабочего стола достаточно «перетащить» его к мусорному ведру.

Контекстное меню стола. Щелчок правой кнопкой по пустому месту на рабочем столе показывает меню с различными вариантами. Следует выбрать Создать Папку, чтобы создать новую папку, или Создать Документ, чтобы создать новый документ.

Панели. При первом запуске рабочий стол GNOME содержит две панели, первая из которых расположена сверху экрана (рис. 3.43, 2), а вторая — у его основания (рис. 3.43, 3). Первая включает три меню — Приложения (рис. 3.44, 1), Переход (рис. 3.44, 2), область быстрого запуска (рис. 3.44, 3), содержащая кнопки для часто используемых программ (Web-браузер Firefox, редактор OpenOffice.org и пр.), а также кнопку выхода из сеанса. Здесь же размещен системный лоток (рис. 3.44, 4), содержащий изображения апплетов (SUSEWatcher, SUSEPluggger и пр.) и контрольная область с часами и управлением уровнем громкости аудиовоспроизведения (рис. 3.44, 5).



Рис. 3.44. Верхняя панель рабочего стола GNOME:

1 — меню Приложения; 2 — Переход; 3 — панель быстрого запуска приложений;
4 — системный лоток; 5 — контрольная область

Приложения — открывает легко доступную иерархию приложений, установленных на ПЭВМ, большинство из которых сгруппированы в подменю, соответствующие различным категориям (Система, Офис, Интернет и пр.).

Переход — обеспечивает перемещение к типичным местоположениям наподобие домашнего каталога, дисковых накопителей, рабочего стола и папок в сети. Процедура поиска документов или файлов может также быть запущена из этого меню.

Щелчок правой кнопкой мыши по пустому месту панели открывает меню, предлагающее помощь, информацию и команды для GNOME и панели. Если выбрать пункт Свойства, будет открыт диалог конфигурирования, где можно изменить положение и фон панели. Если выбрать Добавить к панели, то могут быть добавлены кнопки запуска приложений, инструменты, различные апплеты к существующим панелям, а также может быть удалена либо добавлена новая панель.

Нижняя панель слева содержит кнопки с наименованиями всех активных приложений (панель задач, рис. 3.45, 1). Если нажать указателем мыши по выбранной кнопке, то окно приложения перемещается на передний план. Справа на нижней панели расположен переключатель рабочих столов (рис. 3.45, 2).

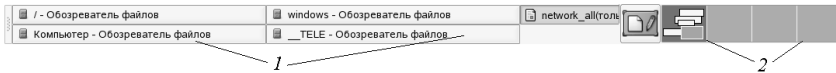


Рис. 3.45. Нижняя панель рабочего стола GNOME:
1 — панель задач; 2 — переключатель рабочих столов

Nautilus — менеджер файлов и Web-обозреватель GNOME. Он может быть использован для создания, просмотра и управления папками, документами и файлами, запуска скриптов, записи данных на CD и установления связи с URL в Internet.

На рис. 3.43, 4 показана последовательность экранов, соответствующая навигации по уровням данных — Компьютер\Файловая система\Windows\D, в результате которой можно найти и просмотреть оглавление диска D:. В конечном итоге найден и открыт файл MS Word — network_all.doc на диске D: (рис. 3.43, 5).

Задания

1. Использование команд операционной системы Unix.

Перейдите в каталог `/etc`.

Укажите шаблоны, которые подходят для следующих имен файлов:

- всех имен;
- всех имен, которые начинаются с `.`;
- всех имен, которые начинаются с `.c`;
- всех имен, которые начинаются с `a` и оканчиваются на `.f`;
- всех имен, которые оканчиваются не на `v`, `w`, `x`, `y` или `z`;
- всех имен, которые состоят из трех строчных букв;
- всех имен, которые содержат, по крайней мере, одну гласную букву;
- всех элементов, которые начинаются с буквы и имеют длину в два символа.

Что происходит, если shell не находит подходящего элемента?

2. Панель Kicker.

- добавить к панели кнопку Недавние документы;
- проверить действие кнопки;

- удалить кнопку с панели;
- добавить апплет Двоичные часы;
- настроить вид Двоичных часов;
- удалить апплет с панели;
- добавить на панель приложение Office Suite;
- открыть и отредактировать текстовый документ;
- открыть и отредактировать электронную таблицу;
- удалить кнопку Office Suite;
- добавить приложение из каталога `/bin` или `/usr/bin` (например, редактор `vi`). Установить режим запуска в окне терминала;
- запустить приложение (`vi`);
- удалить приложение с панели.

3. Управление окнами приложений.

- с помощью функции поиск файла найти на диске все файлы типа `.ps` (PostScript);
- в окне Терминал запустить программу KGhostView;
- просмотреть один из найденных файлов программой KGhostView;
- то же повторить для файлов типа `.pdf`;
- изменить панель инструментов KGhostView (значки — крупные, подпись под значком, размещение панели — справа в окне);
- восстановить вид панели инструментов.

Глава 4

СРЕДЫ И ОБОЛОЧКИ ОПЕРАЦИОННЫХ СИСТЕМ

Понятие оболочки операционной системы возникло в связи с расширением функций первой из массовых ОС — OS/360 (1964 г.). Эта классическая операционная система первоначально была рассчитана на пакетную обработку заданий, поскольку технологии тех времен не могли предложить в качестве терминала ничего лучше электрической пишущей машинки. Появление первых алфавитно-цифровых терминалов (1971 г.) вызвало к жизни широкий перечень диалоговых систем обрамления OS/360/370/375 — CRJE (Control Remote Job Entry — Диалоговый удаленный ввод заданий), CICS (Customer Information Control System — система управления использованием информации) и другие средства. Кроме того, наиболее популярные средства наблюдения и контроля за вычислительным процессом включали в свой состав текстовые редакторы и частично перекрывали функции утилит операционной системы.

Безусловно, к оболочкам ОС следовало бы также отнести и такие средства, как Windows 3.1, GNOME и KDE, рассмотренные выше. Мы тем не менее оставляем их там, где они уже есть, а в настоящей главе ограничимся рассмотрением программных средств, предназначенных для поддержки пользовательского интерфейса с ОС, реализующих команды, приведенные в табл. 1.4, и ряд родственных функций.

4.1. Диалоговые мониторы ЕС ЭВМ

В свое время было известно большое число разновидностей диалоговых мониторов (JEC, RIM, PRIMUS, FOCUS и др.) отечественной разработки, поскольку существовали различные подходы к компенсации недостатков средств ОС ЕС. Наиболее суще-

ственным из этих недостатков являлось отчуждение пользователя от вычислительного процесса, заложенное в разделении труда оператора ЭВМ и программиста, являющееся основным принципом ОС ЕС ЭВМ.

Вот некоторые принципы организации и функционирования диалоговых мониторов на примере системы PRIMUS.

Система PRIMUS выделяет каждому пользователю терминал и закрепляет за ним на диске рабочий набор данных (РНД), выполняющий функции буфера для формирования и редактирования текстов (исходных данных, программ и заданий).

Пользователь общается с системой на двух уровнях: командном и подкомандном. Команды (или функции) вызываются вводом идентификатора команды (обычно 4-символьного) в ответ на подсказку ПОСЫЛАЙТЕ КОМАНДУ. Идентификатор команды обычно сопровождается аргументами, наиболее употребительными из которых являются имя набора данных и название тома. Большинство команд предполагает использование подкоманд, как правило, односимвольных; в этом случае экран терминала обычно может быть разбит на три зоны: подкомандная строка, информационная зона, строка меню. Рассмотрим основные функции PRIMUS, соответствующие процессу подготовки и отладки программ.

Основные группы функций

Работа с наборами данных на МД

DSET — создание набора данных (НД) с указанием по запросу системы параметров DSN, VOL, DCB, SPACE.

INPT — заполнение буфера (РНД) вводимыми текстовыми данными (основной режим формирования файлов программ, заданий, данных).

SAVE [имя_файла[,V=имя_тома]] — запись содержимого РНД в личный файл пользователя на МД.

COPY [имя_файла[,V=имя_тома]] имя_файла — копирование личного НД в РНД; если опущен параметр V, подразумевается имя тома, заданное командой DVOL (см. ниже).

CORR — переход в режим редактирования содержимого РНД.

Основные подкоманды CORR:

D — удаление строки или группы строк;

I — вставка строки или группы строк;

R — замена символов.

В данном режиме используется функциональная часть клавиатуры для вставки/удаления символов и других операций: <F12> — вставка символа; <F10> — удаление символа и т. д.

F 'текст' — контекстный поиск, позволяющий локализовать строку НД, содержащую текст;

C 'текст1', 'текст2' — контекстная замена, локализуя текст1 и заменяющая его в РНД на текст2;

G X — переход к строке РНД номер X;

H — переход в режим просмотра и редактирования в шестнадцатеричном формате представления данных;

E — выход из редактора;

FROM — команда копирования в РНД фрагментов из личных НД;

PRINT [имя_файла[, V=имя_тома]] — распечатка РНД или личного НД;

LOOK [файл[, V=том]] — просмотр набора данных «на месте», без копирования в РНД. Допускается редактирование в режиме подкоманды R (вставки и удаления строк не разрешены).

Работа с томами и библиотеками

CONT [файл[, V=том]] — просмотр содержимого библиотечного НД, состоящего из именованных разделов. Разделы библиотеки, имена которых выводятся на экран, могут быть подвергнуты обработке в соответствии со следующими подкомандами:

* — просмотр раздела (см. LOOK);

— печать раздела (см. PRNT);

— — удаление раздела;

+ — переименование раздела;

< — копирование раздела в РНД (см. COPY);

E — окончание работы.

VTOS [метка_тома] — просмотр оглавления тома МД.

Подкоманды:

* — просмотр НД (или оглавления библиотечного НД);

— — удаление НД;

+ — переименование НД;

E — окончание работы с командой.

Экран, выводимый командой VTOS, представляет собой таблицу, содержащую следующие сведения о НД: DSNAME, DCB, SPACE, фактически занятое пространство.

Работа с заданиями

START, EXEC — запуск задания, JCL-операторы которого находятся в РНД или личном НД на МД;

D A — просмотр списка активных заданий;
D N — просмотр входных/выходных очередей;
CONS — просмотр копии экрана главной дисплей-консоли оператора ЭВМ;

OPER — переход в режим консоли оператора ЭВМ, позволяющий пользователю не только наблюдать за вычислительным процессом, но и активно вмешиваться в его ход;

SOUT имя_задания — просмотр результатов работы заданий.

Подкоманды:

L X — выбор для просмотра одного из нескольких выходных файлов задания;

H [X] — вывод файла (или всех результатов) на печать;

C — уничтожение результатов в выходной очереди.

Некоторые специальные команды

DASD — просмотр имен и адресов накопителей на МД;

DVOL — установка по умолчанию имени текущего НМД;

ACRS — работа с указателем пользователей системы (регистрация), установка/изменение статуса, исключение из пользователей;

CANC — завершение работы системы PRIMUS.

Кроме вышеперечисленных, диалоговая система PRIMUS обеспечивала также функции регистрации пользователей, копирования НД и некоторые другие, частично перекрывая и дополняя возможности утилит ОС ЕС.

4.2. Монитор PCTools для ПЭВМ

PCTools (рис. 4.1) являлся первым из известных средств расширения командного языка MS DOS, предусматривавшим достаточно типовые средства работы с устройствами, файлами, текстами. В дальнейшем был практически вытеснен программным средством Norton Commander.

Запуск PCTools

Для запуска ptools надо ввести команду pct с клавиатуры. Если программу запустить с ключом pct /RnnnK, программа становится резидентной в основной оперативной памяти и мо-

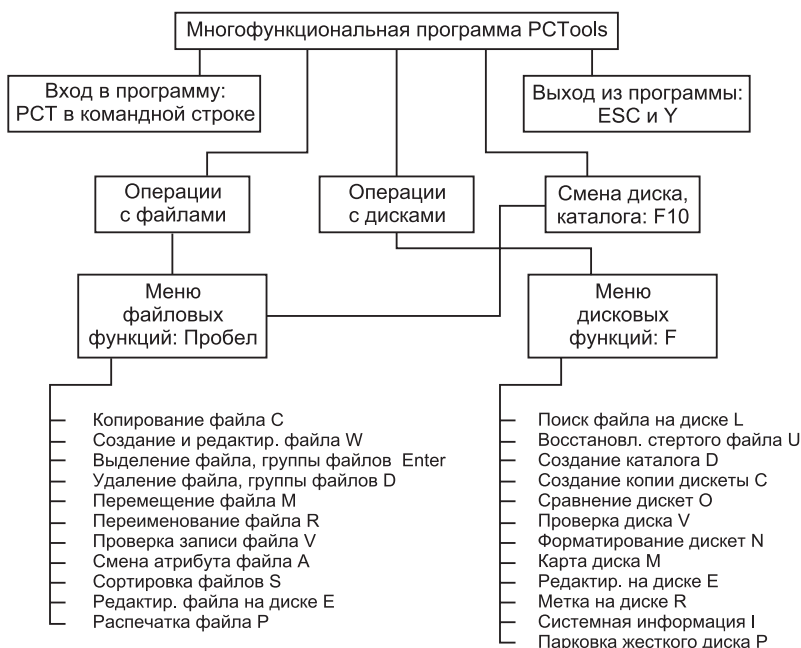


Рис. 4.1. Основные функции PCTools

жет быть вызвана в любой момент нажатием комбинации клавиш <Ctrl+Esc>. Вместо *nnn* следует указать объем памяти (которую будет занимать программа в килобайтах), который не должен быть меньше 64 Кбайт. При этом на диске в корневом каталоге будет создан файл `pctools.ovl`.

Для выхода из PCTools надо нажать <Esc> и подтвердить выход Y. После запуска программа предлагает нажать <F3> для перехода к меню дисковых функций или любую другую клавишу для перехода к меню файловых функций. Для смены диска или каталога надо нажать <F10>. Соответствующие функции выбираются нажатием выделенной клавиши. После выбора функции обычно появляется уточняющее меню, в котором с помощью стрелок следует выбрать режимы работы. Нажатие клавиши <Esc> в уточняющем меню вызывает аварийное прерывание функции и выход в меню предыдущего уровня. Используя клавиши <Enter> и <F1>, можно выбрать несколько файлов из представленных на экране, и тогда функции будут выполняться со всеми выделенными файлами. Для получения

помощи в меню файловых или дисковых функций следует нажать клавишу <N>. Программа PCTools при относительно небольшом собственном длине выполняет такое же множество функций, как Norton Commander, плюс многие функции из Norton Utilities. Эту программу полезно иметь на системных и ремонтных дисках.

Средство PCTools представляет пользователю два основных режима работы:

- файловые функции (работа на логическом уровне) — рис. 4.2;
- специальные функции (работа на физическом уровне) — рис. 4.3.

Advanced PC Tools 5.0					Vol Label=REANIMATOR				
File Functions					Scroll Lock OFF				
Path=A:*.*									
Name	Ext	Size	Attr	Date	Name	Ext	Size	Attr	Date
TMM_IPS	DOC	125952	...A	11/28/01					
BOOK	DOC	525824	...A	3/05/02					
Ф3M_CT~1	RTF	82589	...A	1/21/02					
Ф3M_CT~2	RTF	107666	...A	1/21/02					
4 files LISTed		=	842031 bytes.	4 files in sub-dir		=	842031 bytes.		
0 files SELECTed		=	0 bytes.	Available on volume		=	614912 bytes.		
Copy Move cOmp Find Rename Delete Ver view/Edit Attrib Wordp Print List									
Sort Help ^L=SELECI F2=alt dir lst F3=other menu Esc=exit PC tools									
F8=directory LIST argument F9=file SELECTION argument F10=chg drive/path									

Рис. 4.2. Режим работы с файлами PCTools

Advanced PC Tools 5.0																	Disk View/Edit Service																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
Path=A:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
Absolute sector 00000000, System BOOT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
Displacement																	Hex codes																	ASCII value																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
0000(0000)	EB	3C	90	2A	49	4A	4D	72	49	48	43	00	02	01	01	00	00	01	61	J	m	+	I	H	C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0</

Рис. 4.3. Режим специальных функций PCTools

Файловые функции

Программа PCTools выполняет следующие операции с файлами в меню файловых функций (рис. 4.2): создание и редактирование файла W, удаление файла D, перемещение файла M, переименование файла R, копирование файла на диск или на дискету C, проверка записи файла на диске V, смена атрибута файла A, сортировка по имени, расширению, размеру, времени создания файла S, редактирование файла прямо на диске E, распечатка файла на принтере P и др.

Для выполнения этих функций следует нажать на клавишу с указанной буквой в данном меню.

Дисковые функции

Программа PCTools производит следующие операции с файлами, каталогами и дисками в меню дисковых функций (рис. 4.3, <F3>): поиск файлов на диске по имени или расширению L, восстановление удаленных файлов U, создание, переименование, перемещение и смена каталогов D, создание точных копий дискет C, сравнение дискет O, проверка диска на наличие физических дефектов и отметка поврежденных кластеров V, форматирование дискет N, вывод на экран карты диска или дискеты M, просмотр и редактирование информации прямо на диске E, создание и смена метки диска или дискеты R, выдача системной информации о компьютере I, парковка жесткого диска P и др. Для выполнения этих функций надо нажать на клавишу с указанной буквой в данном меню.

4.3. Оболочка NORTON COMMANDER (DOS) и ее графические аналоги для Windows

Оболочка Norton Commander (NC) являлась самой распространенной из надстроек над DOS, преобразующих ее командный пользовательский интерфейс в интерфейс типа «меню». Она стала настолько привычной для пользователей ПК, что с ней не хотят расставаться даже те из них, кто уже давно работают в среде Windows, применяя оболочки Norton Commander в качестве «фай-

лового манипулятора». Причины привязанности многих к «классической» оболочке Norton Commander в ее исключительной простоте, привычности работы с ней, в экономном использовании ею ресурсов ПК. Оболочка Norton Commander стала настолько неотъемлемым, естественным атрибутом IBM-совместимого ПК, что всякий, кто по необходимости или любознательности знакомится с новой оболочкой, невольно проецирует ее возможности на возможности оболочки Norton Commander. Продолжением оболочки для сред Windows являются Windows Commander, Norton for Windows, FAR Manager.

Оболочка Norton Commander разработана американской фирмой Peter Norton Computing, которая с 1990 г. входит в состав корпорации Symantec.

Рассмотрим здесь лишь вкратце основные возможности и приемы работы с NC, поскольку ниже будет подробно описан и разобран «нортонообразный» продукт для ОС Windows 95/98/NT/2000/XP — Far Manager.

Основные возможности оболочки

Оболочка Norton Commander обеспечивает:

- отображение деревьев каталогов и содержимого каталогов (характеристик входящих в них файлов) в форме, наиболее удобной для восприятия человеком;
- выполнение всевозможных действий с каталогами, файлами и целыми поддеревьями файловых структур, включая их создание, копирование, пересылку, переименование, удаление и поиск, а также смену атрибутов файлов;
- в максимальной степени естественную работу с архивами, включая отображение их содержимого, а также создание, обновление и распаковку архивов (архив представляет собой файл, в котором находится группа сжатых по специальному алгоритму файлов);
- визуализацию файлов, подготовленных популярными текстовыми и графическими редакторами, системами управления базами данных, электронными таблицами и другими прикладными программами;
- подготовку текстовых файлов;
- выполнение из ее среды практически всех команд DOS;

- запуск программ, для чего используются различные, наиболее удобные для пользователя способы;
- выдачу информации о компьютере в целом, о дисках и об оперативной памяти;
- поддержку межкомпьютерной связи через последовательный или параллельный порт (с использованием модуля Commander Link);
- поддержку электронной почты через модем по телефонным линиям связи (с помощью модуля Term90, разработанного для корпорации Symantec фирмой BAUSCH datacom GmbH).

Оболочка Norton Commander, как и любая другая оболочка, упрощая взаимодействие пользователя с ПК, полностью все же не освобождает его от необходимости знать пользовательский интерфейс DOS, так как многие функции доступны только на уровне системы или реализуются на этом уровне гораздо эффективнее.

Оболочка Norton Commander столь привлекательна не в последнюю очередь благодаря великолепным высокоскоростным средствам визуализации данных и развитым средствам электронной почты.

Визуализация файла состоит в форматировании его содержания (в подготовке для вывода на экран в форме, пригодной для восприятия) с последующим отображением результата на экране монитора. Формат файла распознается оболочкой автоматически, исходя из расширения его имени и, при необходимости, внутренней структуры.

К достоинствам рассматриваемой оболочки относятся:

- высокая степень интеграции функций;
- удобство выдачи команд DOS — выдавать их из среды оболочки даже удобнее, нежели взаимодействуя с DOS непосредственно;
- поддержка иерархической системы меню (вложенных меню) для запуска программ;
- простота освоения и удобство использования;
- высокая устойчивость в работе и приемлемая защищенность от ошибок пользователя;
- наличие удобного и понятного контекстно-чувствительного интерактивного справочника;
- поддержка манипулятора типа мышь.

Наряду с неоспоримыми достоинствами имеются и некоторые недостатки. Среди них:

- отсутствие средств сортировки каталогов в дереве файловой структуры;
- невозможность выполнения групповых операций над файлами в различных каталогах, не говоря уже о файлах на различных дисках;
- невозможность выполнения каких-либо действий с группами файлов, найденными на диске средствами оболочки (начиная с версии 4.0 выполнение действий с отдельными файлами возможно).

Оболочка Norton Commander не предъявляет к оборудованию ПК никаких особых требований.

Для размещения всех файлов, образующих оболочку, требуется около 1,8 Мбайт дискового пространства. Основные составляющие файлы пакета: `nc.exe` — загрузчик, `nc.cfg` — описание конфигурации оболочки, `nc.hlp` — справка, `ncedit.exe` — редактор текстов, `ncmain.exe` — файловый менеджер, `nczip.exe` — упаковщик, `packer.exe` — программа-интерпретатор упаковщика, `ncff.exe` — поиск файлов, `ncclean.exe` — очистка дисков, `ncdd.exe` — копирование дискет, `nclabel.exe` — изменение метки диска, `ncnet.exe` — работа в локальной сети, `ncsf.exe` — форматирование дискет, `ncsi.exe` — просмотр системной информации, `*.msg` — сообщения соответствующей утилиты, специальной программы просмотра, гашения экрана, прослушивания wav-файлов.

Norton Commander запускается файлом `nc.exe` (либо `ncsmall.exe`, `ncmain.exe`). После запуска программы высвечивает на экране в так называемом «окне» содержимое активного каталога.

Типовая конфигурация экрана NC была приведена ранее (см. рис. 1.27), и здесь мы отметим лишь, что она включает:

- панели (левую и правую) для отображения информации о файлах, дисках и пр., а также для манипулирования файлами и программами;
- ниспадающее (свешивающееся, pull-down) меню, предназначенное для управления видом панелей, конфигурирования системы и др. функций;
- зону командной строки и протокола работы;
- меню использования функциональной клавиатуры.

Для передвижения по «окну» можно использовать стандартные клавиши управления положением курсора (стрелки <←→↑↓>, <Del>, <Ins>, <Home>, <End>, <PgUp>, <PgDn>).

Стрелки служат для перемещения курсора в активном «окне». Клавиши <PgUp> и <PgDn> служат для постраничного перемещения по тексту соответственно вверх и вниз (перелистывание экранных страниц). Клавиши <Home> и <End> предназначаются для перемещения курсора в начало и конец строки соответственно. Клавиша <Ins> выполняет подсветку (выбор) программ для последующего копирования, удаления и т. д. Для отмены подсветки достаточно повторно нажать на <Ins>. Работает только с именами файлов и не работает с именами подкаталогов.

Клавиша <Del> служит для удаления символа, на котором находится курсор. Последний при этом остается на прежнем месте, а символы справа от курсора сдвигаются на одну позицию влево. Для перехода в другой подкаталог достаточно подвести «подсветку» (highlight) к имени подкаталога (подкаталоги изображаются в окне прописными буквами) и нажать клавишу <Enter>. После этого в окне будет отображено содержимое данного подкаталога. Содержимое второго окна останется без изменения — это окно *не активно*. Для смены активного окна (т. е. окна, в котором производится работа) достаточно нажать клавишу <Tab> или <Ctrl+I>. Эта операция меняет активное окно. Для возврата в подкаталог предыдущего уровня надо подвести подсветку к имени подкаталога предыдущего уровня и нажать <Enter> или <Ctrl+PgUp> (первый способ не срабатывает, если командная строка не пуста, второй же способ игнорирует заполненность командной строки).

Norton Commander позволяет также запустить любые заранее выбранные программы, используя специальное пользовательское меню. Для этого достаточно нажать функциональную клавишу <F2> и выбрать из предлагаемого списка программу.

Операции над файлами

Выделение файлов и каталогов. Выделение файлов и каталогов используется, как правило, непосредственно перед их копированием, перемещением или удалением.

1. Для выделения (отмены выделения) малых групп или одиночных файлов и каталогов нужно использовать клавишу <Ins> или правую кнопку мыши.

2. Для выделения (отмены выделения) всех файлов в каталоге удобно использовать клавишу <*>, расположенную справа. Для выделения всех файлов также можно использовать комбинацию клавиш <++Enter>, для отмены выделения — <--Enter>. Клавиши <+> и <-> расположены справа. Клавишу <*> можно использовать для инверсии выделения.

3. Для выделения группы файлов с одинаковым расширением нужно нажать клавишу <+>, расположенную справа, затем ввести в окно Выбор файлов вместо правой звездочки расширение файла, например *.txt, затем нажать <Enter>. Для отмены такого выделения использовать <--Enter>. Аналогично выделяют группу файлов с одинаковым именем, например nc.* в каталоге NC.

Копирование файлов и каталогов

1. Открыть каталог с файлами и (или) подкаталогами.

2. Осуществить выделение файлов и каталогов, предназначенных для копирования. Если копируется один файл или один каталог, то на него достаточно поместить курсор.

3. На противоположной панели открыть каталог для приема файлов и подкаталогов или вставить дискету и открыть ее (<Alt+F1> или <Alt+F2>), если копирование производится на дискету.

4. При копировании курсор должен находиться на стороне копируемых файлов или каталогов.

При копировании файлов с дискеты на диск курсор должен находиться со стороны выделенных файлов на дискете, а на диске следует открыть каталог для приема файлов.

5. Нажать клавишу <F5> и <Enter>, если копируются только файлы. При копировании нескольких подкаталогов после нажатия <F5> надо указать мышью Включая подкаталоги и затем <Enter>. Прерывание или отмена копирования — <Esc>.

Перенос (перемещение) файлов и каталогов. Отличается от копирования тем, что вместо <F5> надо нажать <F6>.

Переименование файлов и каталогов

1. Установить курсор на файл или каталог.

2. Нажать клавишу <F6> и очистить клавишей <Del> или <BS> окно Переименование файлов.

3. Ввести в окно Переименование файлов новое имя (расширение) файла или каталога и нажать <Enter>.

Удаление файлов и каталогов

1. Выделить подлежащие удалению файлы и каталоги и установить курсор со стороны выделенных файлов (каталогов).

2. Нажать клавишу <F8>, появится окно Удаление файлов и затем нажать <Enter>, если удаляются только файлы. В появившемся втором окне подтверждения удаления указать Все и <Enter>. При удалении одного файла с атрибутом Архивный окно подтверждения удаления не появляется.

При удалении нескольких подкаталогов после нажатия <F8> надо указать мышью Включая подкаталоги и затем <Enter>. В появившемся втором окне подтверждения удаления указать Все и <Enter>. Отмена удаления — <Esc>.

Поиск файлов на диске

1. Нажать <Alt+F7>. Появляется окно Поиск файла. Отмечаем, если не отмечено. Место поиска — Весь диск C: или Каталоги (вводим имя каталога для поиска). Диск при необходимости можно сменить кнопкой Диск, если поиск проводится не на диске C:.

2. Ввести в окно Найти файлы имя файла или его часть и (или) расширение файла или его часть. Недостающие части имени или расширения заменить знаками <*>, когда число недостающих букв неизвестно, или знаками <?> вместо недостающих букв, когда число недостающих букв известно.

3. Указать мышью кнопку Старт. Повторный поиск — кнопка Новый.

4. Выход из режима поиска файлов — <Esc>.

Поиск каталогов на диске

1. После нажатии <Alt+F10> появляется окно Дерево каталогов. При первом поиске создается файл treeinfo.ncd со списком всех каталогов на диске C:.

2. В окно Поиск ввести первую букву или цифру имени каталога. Если найденный каталог не тот, что нужен, то вводить вторую букву имени и т. д., пока не будет найден искомый каталог. Если буква или цифра не вводится, значит такой каталог отсутствует.

3. Чтобы проверить, нет ли других каталогов с тем же именем, следует нажать <Ctrl+Enter> несколько раз, пока не будут найдены все каталоги.

4. Для перехода в искомый каталог необходимо нажать <Enter>, когда курсор установлен на нем в результате поиска.

5. Выход из режима поиска каталога — <Esc>.

Быстрый поиск файла в известном каталоге

1. Поиск файлов будет успешней и быстрее, если предварительно установить сортировку файлов по имени (<Ctrl+F3>). Открыть каталог с файлами. Если в каталоге есть подкаталоги, то установить курсор на первый файл.

2. Нажать клавишу <Alt> и ввести первую букву (цифру) в имени файла. Курсор переместится на файл, начинающийся с введенной буквы. Если файла на эту букву нет, то буква не будет вводиться. Продолжать ввод при необходимости следующей буквы имени файла, пока не будет найден искомый файл.

Смена атрибута файла. Файл может иметь один из следующих четырех атрибутов:

1. Архивный (Archive) — это основной атрибут файлов. При создании файл, как правило, по умолчанию получает этот атрибут.

2. Только для чтения (Read Only) — применяется для защиты файла от случайного удаления. При удалении такого файла будет выдан запрос на удаление. Архивный файл удаляется без запроса.

3. Скрытый (Hidden) — применяется для защиты от удаления особо важных файлов.

4. Системный (System) — аналогичен скрытому файлу.

Для смены атрибута файла используют Меню\Файл\Атрибут, выбирают мышью или пробелом один из указанных атрибутов и устанавливают его (Set). Для смены атрибута одновременно у целой группы файлов ее выделяют, затем Меню\Файл\Атрибут, устанавливают новый атрибут (Set) и удаляют старый атрибут (Clear).

Для того чтобы сделать видимыми (или невидимыми) скрытые и системные файлы, необходимо вызвать Меню <F9>, пункт Option\Configuration\Show Hidden Files, указать OK.

Действие Norton Commander при нажатии клавиши <Enter>

1. Если выделено имя каталога, то при нажатии <Enter> будет выведено оглавление каталога.

2. Если курсор установлен на файл с расширением exe, com, bat, то при нажатии <Enter> начнется исполнение этого файла (запуск программы).

3. Если выделено имя архива (файлы с расширением arj, zip, lzh, arc, pak, ice, zoo), то при нажатии <Enter> Norton

Commander войдет в архив и выведет оглавление архива. Это свойство NC используется для архивации.

4. Если выделенный файл имеет расширение, указанное в файле `nc.ext`, то действие Norton Commander при нажатии `<Enter>` будет определяться содержанием файла `nc.ext`.

Другие программные оболочки

Наиболее распространенными оболочками, унаследовавшими основные черты Norton Commander, являются, по-видимому, Windows Commander, Norton for Windows, FAR Manager. Ниже кратко перечислены основные особенности первых двух оболочек, после чего мы подробно остановимся на Far Manager, которая является отечественной разработкой.

Norton for Windows (рис. 4.4), как это очевидно, использует элементы интерфейса Windows — кнопки, линии прокрутки и пр. Основные особенности:

- поддержка 32-разрядной архитектуры;
- поддерживаются все программы просмотра файлов, характерные для Windows;

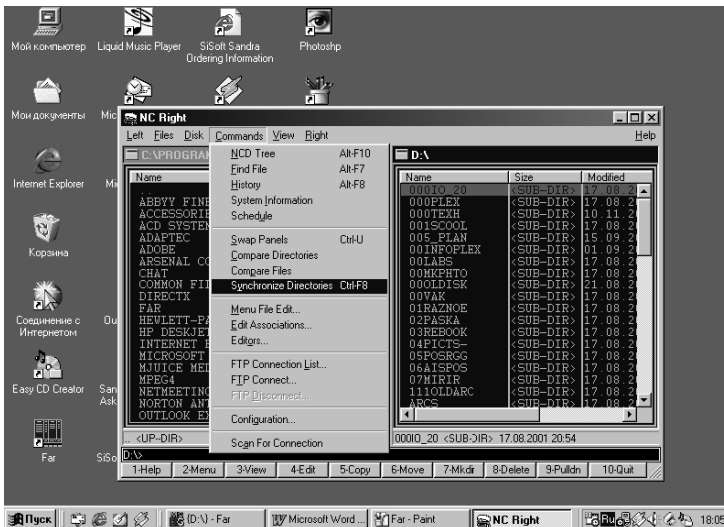


Рис. 4.4. Типичный экран NC for Windows в среде Windows. Можно видеть панели файлов и меню Commands

- реализована улучшенная функция перемещения файлов и каталогов (буксировка) из разных окон, в том числе и с рабочего стола Windows;
- поддержка длинных имен файлов (LFN);
- улучшенная система меню;
- реализована контекстная подсказка по правой кнопке мыши;
- включена программа-клиент FTP для легкой передачи файлов по данному протоколу.

Windows Commander (см. рис. 4.5—4.8) представляет собой средство манипулирования файлами наподобие Winfile.exe (манипулятор файлами Windows). Однако он унаследовал все основные черты NC — две панели, меню, команды. Основные возможности:

- поддержка функций перемещения файлов с помощью мыши (буксировка), включая и вывод на печать;
- копирование, перемещение, переименование, удаление «деревьев каталогов», включая удаление непустых каталогов;
- обработка архивов как субдиректорий — автоматически подключаются архиваторы pkzip, arj, lha, rar, us2 и ace.

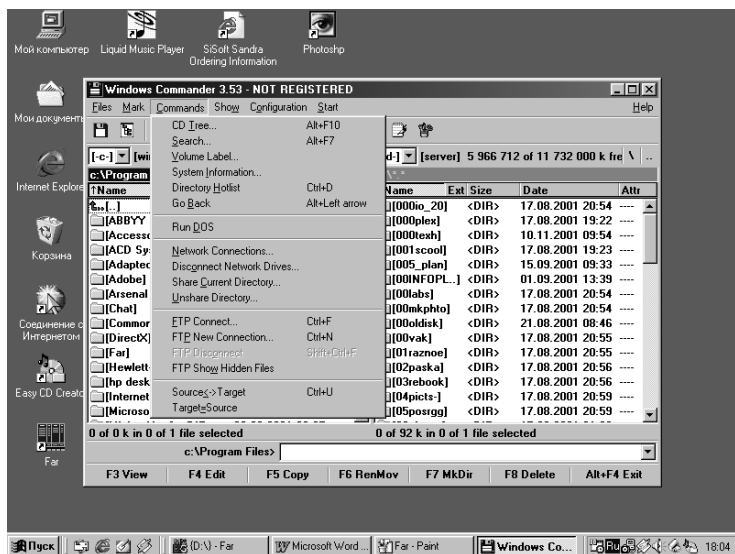


Рис. 4.5. Типичный экран Windows Commander в среде Windows. Видны панели файлов и меню Commands

Встроенная распаковка архивов форматов arj, lzh, gz и tar. Встроенная упаковка архивов zip на основе метода ZLib (Jean-loup Gailly);

- встроенный просмотр (<F3>) файлов любой длины в шестнадцатеричном, двоичном коде, а также ASCII (DOS) или ANSI (Windows);
- конфигурирование запуска внешних программ или внутренних команд;
- поддержка FTP-клиента и т. д.

На рис. 4.6 как раз отображен экран настройки на связь с FTP-сервером. Конфигурации каждой настройки запоминаются в меню Connection и включают в себя:

- адрес FTP-сервера (здесь — ftp.inion.ru);
- имя пользователя и пароль (здесь — anonymous);
- имя удаленного каталога в файловой системе FTP (здесь — w_inion/irb).



Рис. 4.6. Настройка FTP-клиента Windows Commander на связь с сервером

После установления связи на одной из панелей отображается удаленный каталог (см. рис. 4.8 — слева удаленный каталог, справа — локальный).

Передача файлов в обе стороны (Upload и Download) осуществляется обычным выделением файлов (директорий) и копированием их по команде <F5> (рис. 4.7).

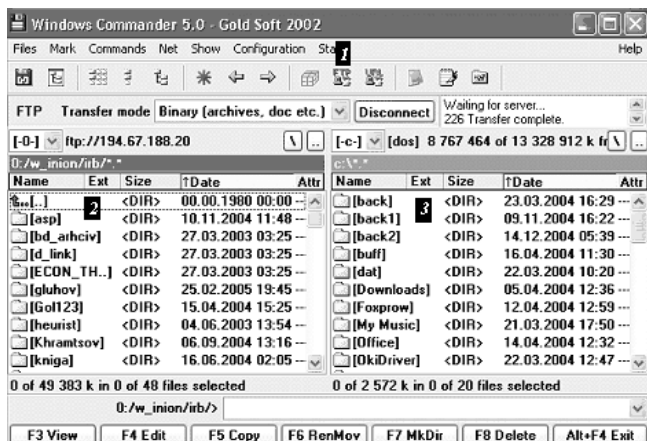


Рис. 4.7. Windows Commander, режим связи с FTP-сервером:

1 — кнопка запуска FTP-клиента; 2 — удаленный каталог; 3 — локальный каталог

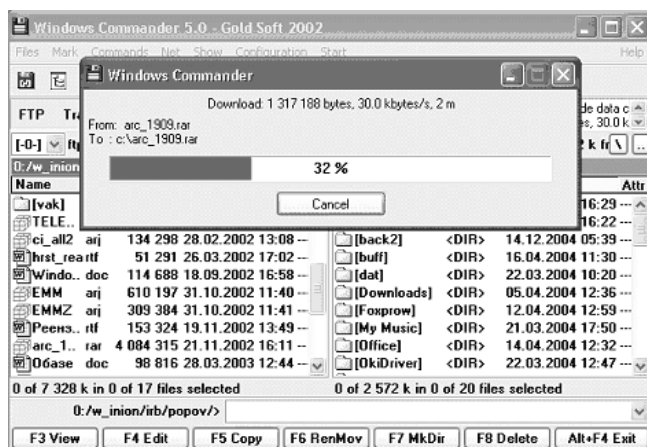


Рис. 4.8. Windows Commander — режим передачи файла (Download) из удаленного каталога в локальный

4.4. FAR Manager — текстовая оболочка для Windows 95/98/NT/2000/XP

FAR — это работающая в текстовом режиме программа управления файлами для Windows с поддержкой *длинных имен файлов* и широким набором операций над файлами и папками.

FAR позволяет работать с архивами. При этом файлы в архивах обрабатываются аналогично файлам в папках. FAR сама преобразует команды в соответствующие вызовы внешних архиваторов. FAR также обеспечивает значительное количество сервисных функций (см. также Приложение 1).

Параметры командной строки

В командной строке при запуске Far Manager могут быть использованы некоторые ключи и параметры (табл. 4.1, табл. П1.1).

Таблица 4.1. Ключи и параметры командной строки Far

Ключи	Назначение	Комментарий
/a	Запрет отображения символов с кодами 0—31 и 255	Может быть полезно при запуске FAR из telnet
/ag	Запрет отображения псевдографических символов	
/e[<строка>[:<позиция>]]<filename>	Редактирование указанного файла	После /e можно дополнительно указать строку и позицию в строке, устанавливаемые после запуска редактора. Например: <code>far /e70:2 readme</code>
/i	Установка маленькой (16 × 16) иконки для окна консоли FAR	В некоторых конфигурациях эта опция может привести к нестабильной работе
/u<username>	Позволяет использование раздельных настроек для различных пользователей	Например: <code>far /u guest</code>
/v<filename>	Просмотр указанного файла	Если в качестве <filename> использовано —, данные будут читаться с stdin. Например, « <code>dir far /v -</code> » выведет результат работы команды <code>dir</code>
Far <filename>	Запуск FAR в режиме просмотра содержимого <filename>, если <filename> — архив, или в режиме просмотра содержимого папки, если <filename> — папка	

Некоторые общие понятия и операции

Прежде чем перейти к описанию команд панелей — основного рабочего инструмента, перечислим некоторые общие понятия.

Маски файлов. Маски файлов часто используются в командах FAR для выбора отдельных файлов и папок или их групп. Маски могут включать обычные допустимые в именах файлов символы, а также специальные выражения:

* — любое количество символов;

? — любой символ;

[c, x — z] — любой символ из находящихся в квадратных скобках. Допускаются и отдельные символы, и их диапазоны.

Например, файлы ftp.exe, fc.exe и f.ext могут быть выбраны с помощью маски f*.ex?, маска *co* выберет и color.ini, и edit.com, маска [c — f, t]*.txt может выбрать config.txt, demo.txt, faq.txt и tips.txt.

Во многих командах FAR можно задать несколько разделенных запятыми масок. Например, чтобы выбрать все документы, можно ввести *.doc, *.txt, *.wri в команде Пометить группу.

Пометка файлов. Для обработки файлов и папок панели файлов они могут быть выбраны несколькими различными способами.

<Ins> помечает файл под курсором и перемещает курсор вниз.

<Shift+Клавиши курсора> позволяют перемещать курсор в различных направлениях.

<Num+> и <Num-> выбирают или снимают пометку с группы с использованием одной или нескольких разделенных запятыми масок файлов.

<Num*> инвертирует текущую пометку.

Команда Восстановить пометку (<Ctrl+M>) восстанавливает выбранную до этого группу.

<Ctrl+Num+> и <Ctrl+Num -> выбирают или снимают пометку со всех файлов с тем же расширением, что и у файла под курсором.

<Alt+Num+> и <Alt+Num -> выбирают или снимают пометку со всех файлов с тем же именем, что и у файла под курсором.

<Ctrl+Num*> инвертирует текущую пометку, включая папки. Если параметр Пометка папок в диалоге Настроек панели включена, это работает аналогично <Num*>.

<Shift+Num+> и <Shift+Num-> выбирают или снимают пометку со всех файлов.

Если ни один файл не выбран, то будет обработан только файл под курсором.

Копирование, перенос, переименование и создание связей. Следующие команды могут быть использованы для копирования, переноса и переименования файлов и папок:

Копировать выбранные файлы — <F5> .

Копировать файл под курсором вне зависимости от пометки — <Shift+F5>.

Переименовать или перенести выбранные файлы — <F6>.

Переименовать или перенести файл под курсором вне зависимости от пометки — <Shift+F6>.

На разделах NTFS также можно создавать жесткие связи файлов с помощью команды <Alt+F6>, так что вы можете иметь несколько различных имен файлов, ссылающихся на одни и те же данные.

Если необходимо создать папку назначения перед копированием, следует добавить к ее имени обратную черту. Также в диалоге копирования можно нажать <F10> для выбора папки из дерева активной файловой панели или <Alt+F10> для выбора из дерева пассивной файловой панели.

Возможность копирования, переноса и переименования файлов для подключаемых модулей зависит от функциональности конкретного модуля.

Если файл, в который производится копирование, уже существует, то он может быть перезаписан, пропущен, либо содержимое копируемого файла может быть дописано в его конец.

Если диск, на который производится копирование или перенос файлов, в ходе операции заполнился, то можно либо отменить операцию, либо заменить диск и выбрать пункт Разделить, после чего копируемый файл будет разделен между дисками. Эта функция доступна только при выключенном параметре Использовать системную функцию копирования из диалога Системные параметры.

Параметр Копировать права доступа может использоваться только для файловой системы NTFS и позволяет копировать информацию о правах доступа к файлу.

Параметр Использовать системную функцию копирования из диалога Системные параметры включает использование функции Windows CopyFileEx (или CopyFile, если CopyFileEx недоступна) вместо внутренней реализации копирования файлов. Это может быть полезно на NTFS, так как CopyFileEx выполняет более эффективное распределение дискового пространства и копирует расширенные атрибуты файлов.

Перетаскивание файлов. Операции копирования и переноса файлов могут быть выполнены с помощью «перетаскивания». Нажмите левую кнопку мыши на исходном файле или папке, перетащите его на другую панель и отпустите кнопку мыши.

Если вы хотите обработать группу файлов или папок, пометьте их перед перетаскиванием, нажмите левую кнопку мыши на исходной панели и перетащите файлы на другую панель.

Вы можете переключаться между копированием и переносом, нажимая правую кнопку мыши во время перетаскивания. Также для переноса файлов вы можете удерживать клавишу <Shift> в момент нажатия левой кнопки мыши.

Меню выбора диска. Это меню позволяет сменить текущий диск панели, отсоединиться от сетевого диска или открыть новую панель подключаемого модуля.

Необходимо выбрать пункт меню с соответствующей буквой диска для смены текущего диска или пункт с названием модуля, чтобы создать новую панель модуля. Если панель не является панелью файлов, ее тип будет изменен на панель файлов.

Для отсоединения от сетевого диска можно использовать <Del>. <Ctrl+1> — <Ctrl+8> переключают отображение различной информации:

<Ctrl+1> — тип диска;

<Ctrl+2> — сетевое имя (и путь, ассоциированный с SUBST диском под NT);

<Ctrl+3> — метка диска;

<Ctrl+4> — файловая система;

<Ctrl+5> — общее и свободное место на диске;

<Ctrl+6> — показ параметров сменных дисков;

<Ctrl+7> — показ имен подключаемых модулей;

<Ctrl+8> — показ параметров компакт-дисков.

Настройки меню выбора диска сохраняются в конфигурации FAR.

Панели

Обычно FAR показывает две панели (левое и правое окна) с различными видами информации (см. также табл. П1.2):

- панель файлов (рис. 4.9, 1);
- панель информации (рис. 4.9, 3);
- панель дерева папок (рис. 4.10, 1);
- панель быстрого просмотра.

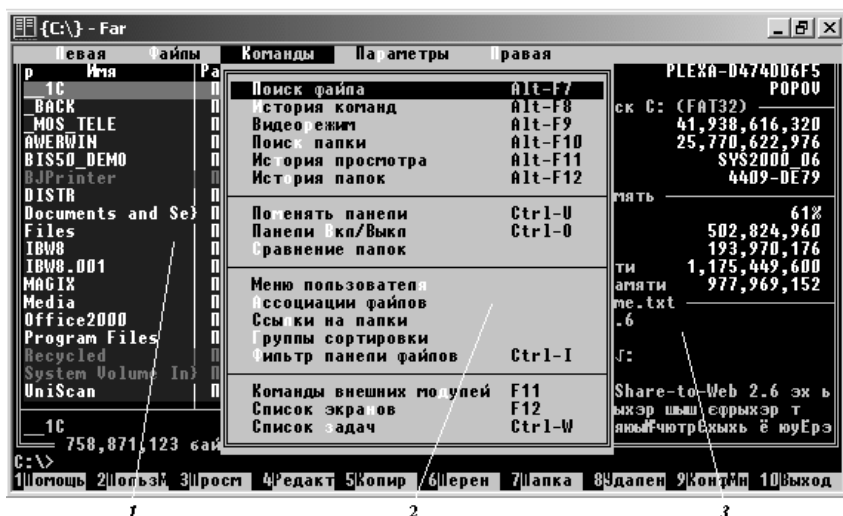


Рис. 4.9. Панель файлов (1), меню команд (2), панель информации (3)

Если необходимо изменить режим просмотра панели, следует выбрать желаемый пункт из Меню Панелей. После смены режима просмотра или текущего диска тип любой панели автоматически меняется на панель файлов.

Панель файлов. Отображает содержимое текущей папки. Вы можете выбирать файлы и папки, выполнять различные файловые и архивные операции. По умолчанию в панели файлов используются следующие режимы просмотра: краткий, средний, полный, широкий, детальный.

Упакованные размеры имеют смысл для файлов с атрибутом Сжатый на дисках с файловой системой NTFS или для файлов внутри архивов. Владельцы и количество связей файлов тоже применимы только к разделам NTFS. Некоторые файловые системы могут не поддерживать время создания и время доступа к файлу.

Для позиционирования на файл можно воспользоваться операцией быстрого поиска по первым буквам имени. Для этого, удерживая клавишу <Alt>, набирайте имя требуемого файла, пока на него не переместится курсор. С помощью <Ctrl+Enter> можно переместиться на следующее имя, соответствующее введенной строке. Кроме обычных символов, в имени файла также можно использовать символы '*' и '?'.

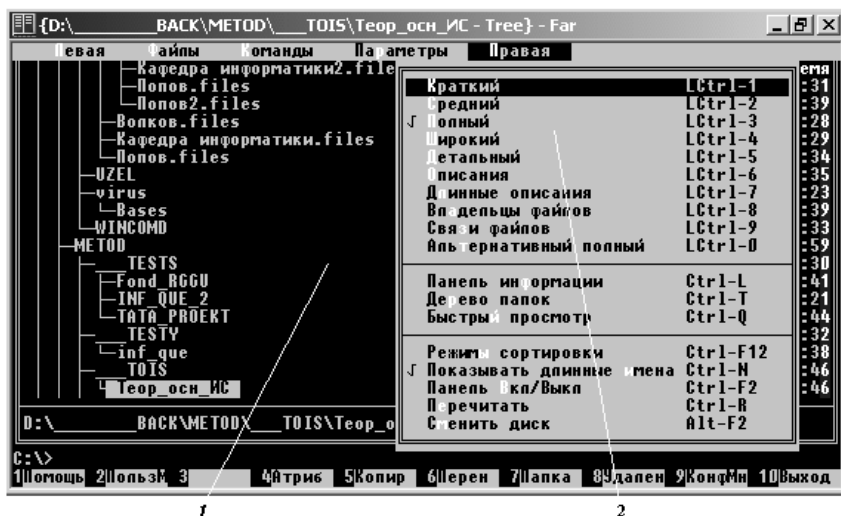


Рис. 4.10. Дерево папок (1) и меню правой панели (2)

Панель дерева папок. Отображает структуру папок текущего диска в виде дерева. Это позволяет быстро сменить текущую папку, а также выполнить операции над папками.

FAR запоминает информацию о структуре папок в файле `Tree.Far`, расположенном в корневой папке каждого диска. Если запись на диск невозможна, то эта информация сохраняется в скрытой папке `Tree.Cache`, находящейся в той же папке, что и `Far.exe`.

Для позиционирования на папку можно использовать операцию быстрого поиска. Для этого, удерживая клавишу `<Alt>`, набирайте имя требуемой папки, пока на нее не переместится курсор. С помощью `<Ctrl+Enter>` можно переместиться на следующее имя, соответствующее введенной строке.

Панель информации. Панель информации содержит следующие данные:

- сетевые имена компьютера и пользователя;
- имя и тип текущего диска, тип его файловой системы, сетевое имя, общий и свободный размеры, метку тома и серийный номер;
- уровень загрузки памяти (100 % означает, что использована вся память), общий и свободный размеры физической и виртуальной памяти;

- файл описания папки, можно просмотреть содержимое этого файла в полноэкранном режиме, нажав <F3> или левую кнопку мыши. Для редактирования или создания этого файла нажать <F4> или правую кнопку мыши.

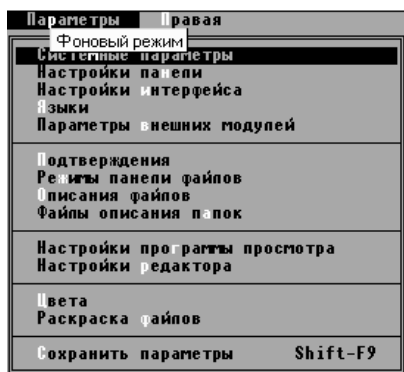
Список возможных имен файлов описания папок может быть задан с помощью команды **Файлы описания папок** в Меню параметров.

Панель быстрого просмотра используется для получения информации о выбранном элементе панели файлов или дерева папок.

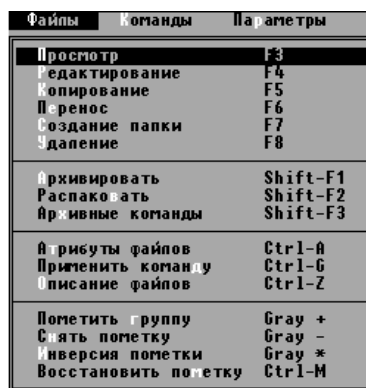
Если выбранный элемент — файл, то отображается его содержимое. Для известных Windows типов файлов также выводится название типа. Для папок в панели быстрого просмотра сообщается общий размер, общий упакованный размер, количество файлов и вложенных папок, размер кластера текущего диска, реальный размер файлов, включая недоиспользованные фрагменты кластеров. Общий упакованный размер применим только для дисков с файловой системой NTFS.

Рассмотрим далее основные рубрики меню Far Manager:

- меню панелей (рис. 4.10, 2);
- меню параметров (рис. 4.11, а);
- меню файлов (рис. 4.11, б);
- меню команд (рис. 4.9, 2).



а



б

Рис. 4.11. Меню параметров (а) и меню файлов (б)

Для активизации меню можно использовать <F9> или нажать кнопку мыши на верхней строке экрана. Комбинация <Shift+F10> позволяет выбрать последний использованный пункт меню.

Меню левой и правой панелей. Меню Левая и Правая позволяют изменить параметры левой и правой панели соответственно. Эти меню включают следующие пункты (рис. 4.10, 2):

- краткий — показывать файлы в три колонки;
- средний — показывать файлы в две колонки;
- полный — показывать имя, размер, дату и время файла;
- широкий — показывать имя и размер файла;
- детальный — показывать имя, размер, упакованный размер, время модификации, создания и доступа и атрибуты файла, полноэкранный режим;
- описания — имя и описание файла;
- длинные описания — имя, размер и описание файла, полноэкранный режим;
- владельцы файлов — имя, размер и владелец файла;
- связи файлов — имя, размер и количество жестких связей файлов;
- альтернативный — имя, размер (форматированный с использованием запятых) и дата создания файла;
- панель информации — сменить панель на панель информации;
- дерево папок — сменить панель на дерево папок;
- быстрый просмотр — сменить панель на панель быстрого просмотра;
- режимы сортировки — показать доступные режимы сортировки;
- показывать длинные имена — показывать длинные/короткие имена;
- панель вкл/выкл — показать/спрятать панель;
- перечитать — перечитать содержимое панели;
- сменить диск — сменить текущий диск.

Меню параметров (рис. 4.11, а). Меню включает:

- системные параметры — вызывает диалог настройки системных параметров;
- настройки панели — вызывает диалог настроек панели;
- настройки интерфейса — вызывает диалог настроек интерфейса;

- языки — выбор основного языка и языка помощи, следует использовать команду «сохранить параметры», чтобы сохранить выбранные языки;
- параметр — настройка параметров подключаемых внешних модулей;
- подтверждения — включение или выключение подтверждений для некоторых операций;
- режимы панели файлов — настройка режимов просмотра панели файлов;
- описания файлов — режимы обновления и имена описаний файлов;
- файлы описания — имена или маски файлов, отображаемых папок в панели информации в качестве описания папки;
- настройки программы просмотра — настройки внешней программы просмотра;
- настройки редактора — настройки внешнего и встроенного редактора;
- цвета — изменение цвета различных элементов интерфейса или изменение всей палитры цветов либо на черно-белую, либо на предлагаемую по умолчанию;
- раскраска файлов — редактирование раскраски файлов;
- сохранить параметры — сохранение текущей конфигурации, цветов и внешнего вида экрана.

Настройка некоторых системных параметров

Подтверждения. В диалоге Подтверждения можно разрешить или запретить подтверждения для следующих операций:

- перезаписывание файлов назначения во время копирования файлов;
- перезаписывание файлов назначения во время переноса файлов;
- перетаскивание (Drag and Drop) файлов;
- удаление файлов;
- удаление папок;
- выход из FAR.

Системные параметры:

- снимать атрибут R/O с CD файлов — снимать атрибут «только для чтения» с файлов, копируемых с CD;

- удалять в корзину — разрешает удаление файлов с использованием корзины (Recycle Bin);
- использовать системную функцию копирования — использовать функцию Windows CopyFileEx (или CopyFile, если CopyFileEx недоступна) вместо внутренней реализации копирования файлов. Это может быть полезно на NTFS, так как CopyFileEx выполняет более эффективное распределение дискового пространства и копирует расширенные атрибуты файлов;
- копировать открытые для записи файлы — позволяет копировать файлы, которые открыты для записи другими программами. Этот режим удобен, когда нужно скопировать открытый в течение долгого времени файл, но он может быть опасным, если этот файл модифицируется одновременно с копированием;
- создавать папки заглавными буквами — если имя новой папки содержит только строчные буквы и установлена эта опция, то папка будет создана заглавными (прописными) буквами;
- время бездействия — завершает работу FAR, если в течение указанного интервала не было нажатий клавиш мыши или клавиатуры, FAR ожидал ввода из командной строки и отсутствовали фоновые экраны редактирования или просмотра;
- сохранять историю команд — вызывает сохранение истории команд перед завершением и ее восстановление после запуска FAR;
- сохранять историю папок — вызывает сохранение истории папок перед завершением и ее восстановление после запуска FAR. Для просмотра содержимого истории папок нажать <Alt+F12>;
- сохранять историю просмотра и редактора — вызывает сохранение истории просмотра и редактора перед завершением и ее восстановление после запуска FAR. Для просмотра входящих в этот список файлов нажать <Alt+F11>;
- использовать стандартные типы файлов — если эта опция включена, то при нажатии <Enter> на файле, тип которого известен Windows и отсутствует в Ассоциациях файлов FAR, будет запущена программа Windows, предназначенная для обработки этого типа файлов;

- автозапись конфигурации — если эта опция включена, FAR будет автоматически сохранять конфигурацию. Также будут сохраняться текущие папки обеих панелей.

Настройки панели:

- показывать скрытые и системные файлы — разрешает показ файлов с атрибутами *скрытый* и *системный*. Этот режим также может быть переключен с помощью <Ctrl+N>;
- раскраска файлов — разрешает раскраску файлов;
- автосмена папки — если эта опция включена, то передвижения курсора по дереву папок будут вызывать смену папки в другой панели. Если эта опция выключена, то для смены папки из дерева папок необходимо нажать <Enter>;
- пометка папок — разрешает пометку папок с использованием <Num+> и <Num*>. В противном случае эти команды работают только с файлами;
- разрешить обратную сортировку — если эта опция включена и текущий режим сортировки файловой панели выбран повторно, то будет установлен режим обратной сортировки;
- показывать заголовки колонок — разрешает показ заголовков колонок панели файлов;
- показывать строку статуса — разрешает показ строки статуса в панели файлов;
- показывать суммарную информацию — разрешает показ суммарной информации в нижней строке панели файлов;
- показывать свободное место — разрешает показ свободного места на текущем диске;
- показывать полосу прокрутки — разрешает показ полосы прокрутки в панели файлов и панели дерева папок;
- показывать количество фоновых экранов — разрешает показ количества фоновых экранов;
- показывать букву режима сортировки — показывать текущий режим сортировки в верхнем левом углу панели;

Раскраска файлов. Диалог Раскраски файлов в Меню параметров позволяет определить группы раскраски файлов. Каждое определение группы включает:

- одну или несколько разделенных запятыми масок файлов;
- атрибуты включения;
- атрибуты исключения;

- цвета обычного имени, помеченного имени, имени под курсором и помеченного имени под курсором. Если вы хотите использовать цвет по умолчанию, установите цвет как Черный на черном;
- опционально может быть указан любой символ для обозначения принадлежащих к группе файлов. Он может быть использован как вместе с цветовым выделением, так и вместо него.

Файл принадлежит к группе раскраски, если:

- его имя соответствует хотя бы одной маске;
- он имеет все атрибуты включения;
- он не имеет атрибутов исключения.

Группы раскраски анализируются от начала к концу. Если обнаружено, что файл принадлежит к какой-либо группе, то принадлежность к остальным группам не проверяется.

Настройка режимов просмотра панели файлов. Панель файлов может выводить информацию, используя 10 заданных заранее режимов (краткий, средний и пр.). Обычно этого достаточно, но тем не менее при желании можно изменить параметры этих режимов или даже полностью заменить их на новые.

Команда Режимы панели файлов из Меню параметров позволяет изменить параметры режимов просмотра. Сначала она предлагает выбрать требуемый режим из списка. В этом списке режим 0 соответствует режиму просмотра, вызываемому по <ЛевыйCtrl+0> (альтернативный полный), режим 1 соответствует краткому режиму (<ЛевыйCtrl+1>), режим 2 соответствует среднему режиму (<ЛевыйCtrl+2>) и т. д. После выбора режима появляется диалог, в котором можно изменить следующие параметры:

- типы колонок — типы колонок кодируются с помощью одной или нескольких букв, разделенных запятыми. Допускаются следующие типы колонок:

$n[m, o, r]$ — имя файла,

где m — показывать символы пометки; o — показывать имена без путей (предназначено в основном для подключаемых модулей); r — выравнивать имена по правому краю (эти символы можно комбинировать, например nmr);

$s[c, t]$ — размер файла,

где c — форматировать размер файла запятыми; t — использовать 1000 вместо 1024 как делитель, если ширины колонки не хватает для показа полного размера файла;

$p[c, t]$ — упакованный размер файла,
где c — форматировать размер файла запятыми; t — использовать 1000 вместо 1024 как делитель, если ширины колонки не хватает для показа полного размера файла;
 d — дата модификации файла;
 t — время модификации файла;
 $dm[b, m]$ — дата и время модификации файла;
 $dc[b, m]$ — дата и время создания файла;
 $da[b, m]$ — дата и время последнего доступа к файлу,
где b — краткий (в стиле Unix) формат времени файла; m — использование текстовых имен месяцев;
 a — атрибуты файла;
 z — описание файла;
 o — владелец файла;
 $l\ n$ — количество жестких связей.

Если описание типов колонок содержит более одной колонки имени файла, панель файлов будет отображаться в многоколоночной форме;

- ширина колонок — позволяет изменить ширину колонок панели. Если ширина равна 0, то используется значение по умолчанию. Если ширина колонки с именем, описанием или владельцем равна 0, она будет подсчитана автоматически в зависимости от ширины панели.

Для правильной работы с различной шириной экрана настоятельно рекомендуется, чтобы в каждом режиме просмотра была хотя бы одна колонка с автоматически вычисляемой шириной.

Для использования 12-часового формата времени надо увеличить на единицу стандартную ширину колонки времени файла или колонки времени и даты файла. После дальнейшего увеличения в этих колонках также будут показаны секунды и миллисекунды.

Для показа года в 4-символьном формате нужно увеличить ширину колонки даты на 2;

- типы колонок строки статуса и Ширина колонок строки статуса — аналогично Типам колонок и Ширине колонок, но для строки статуса панели;
- полноэкранный режим — показывать панель во весь экран вместо половины экрана;
- выравнивать расширения файлов — показывать расширения файлов выровненными;

- показывать папки заглавными буквами — показывать все имена папок в верхнем регистре вне зависимости от реального регистра;
- показывать файлы строчными буквами — показывать все имена файлов в нижнем регистре вне зависимости от реального регистра;
- показывать имена файлов из заглавных букв строчными буквами — показывать все имена файлов, которые содержат только прописные буквы, строчными буквами. По умолчанию эта опция включена.

Все эти параметры влияют только на способ показа файлов, для их обработки FAR всегда использует настоящий регистр;

- использовать регистро-зависимую сортировку — использовать регистро-зависимую сортировку имен файлов.

Настройки интерфейса:

- часы — показывать часы в верхнем правом углу экрана;
- часы при редактировании и просмотре — показывать часы при редактировании и просмотре файлов;
- мышь — использовать мышь;
- показывать линейку клавиш — показывать назначения функциональных клавиш в нижней строке экрана. Эта опция также может переключаться по <Ctrl+B>;
- всегда показывать меню — показывать меню сверху экрана, даже когда оно неактивно;
- сохранение экрана — запуск программы сохранения экрана после заданного в минутах интервала бездействия;
- история в строках ввода диалогов — сохранять историю в строках ввода некоторых диалогов FAR. Список ранее введенных строк может быть вызван с помощью мыши или <Ctrl+↑> и <Ctrl+↓>. Если вы не хотите вести такую историю, например, по соображениям безопасности, выключите эту опцию;
- установить формат командной строки — изменить формат командной строки FAR. Можно использовать следующие переменные:
 - \$p — текущий путь;
 - \$n — буква текущего диска;
 - \$g — символ >;
 - \$s — символ \$;

- использовать правый <Alt> как <AltGr> — установить эту опцию, если имеются проблемы с использованием комбинаций правой <Alt> для ввода символов в Windows 9x, или отключите ее, если вы предпочитаете использовать правую <Alt> для быстрого поиска. Эта опция имеет значение только при работе в Windows 9x и игнорируется под Windows NT;
- показывать общий индикатор копирования — показывать общий индикатор во время выполнения операции копирования. Это может потребовать дополнительного времени перед началом копирования для подсчета общего размера файлов.

Поддержка подключаемых модулей. Внешние подключаемые DLL-модули (plugins) могут быть использованы для создания новых команд FAR и поддержки дополнительных файловых систем. Например, работа с архивами, FTP-клиент, временная панель и просмотр сети реализованы с помощью эмулирующих файловые системы модулей.

Все подключаемые модули хранятся в отдельных папках, размещенных в папке 'Plugins', которая находится в одной папке с Far.exe. При обнаружении нового модуля FAR сохраняет информацию о нем и впоследствии загружает его только при необходимости, так что неиспользуемые модули не требуют дополнительной памяти.

Модули могут быть вызваны либо из Меню выбора диска, либо из меню Команды внешних модулей, активизируемого с помощью <F11> или соответствующего пункта Меню команд. <F4> в меню Команды внешних модулей позволяет назначить горячие клавиши для пунктов этого меню, что упрощает их последующий вызов с помощью клавиатурных макрокоманд. Это меню доступно из файловых панелей и (только по <F11>) из встроенной программы просмотра и редактора.

При вызове из программы просмотра и редактора будут показаны не все модули, а только те, которые специально созданы для работы в этом режиме.

Можно установить параметры модулей, используя команду Параметры внешних модулей из Меню параметров.

Модули имеют собственные файлы сообщений и помощи. Можно получить список доступной помощи по модулям, нажав <Shift+F2> в основной помощи FAR.

Если активная панель отображает файловую систему, поддерживаемую внешним модулем, то команда CD в командной

строке может быть использована для смены текущей папки этой файловой системы. В отличие от `CD`, команда `CHDIR` всегда воспринимает указанный параметр как имя реальной папки вне зависимости от типа файловой панели.

Меню файлов

Меню включает следующие рубрики (см. также табл. П1.3):

- просмотр — просмотр файлов, подсчет размеров папок;
- редактирование — редактирование файлов;
- копирование — копирование файлов и папок;
- перенос — переименование или перенос файлов и папок;
- создание папки — создание новой папки;
- удаление — удаление файлов и папок;
- архивировать — добавить выбранные файлы к архиву;
- распаковать — распаковать выбранные файлы из архива;
- атрибуты файлов — изменить атрибуты и время файла;
- применить команду — применить команду к выбранным файлам;
- описание файлов — добавить описания к выбранным файлам;
- пометить группу — пометить заданную маской группу файлов;
- снять пометку — снять пометку с группы файлов, соответствующей заданной маске;
- инверсия пометки — инвертировать текущую пометку файлов;
- восстановить пометку — восстановить предыдущую пометку после обработки файлов или операции пометки группы.

Установка атрибутов файлов. Эта команда позволяет изменить атрибуты и время как у отдельных файлов, так и у групп файлов и папок. Если вам не нужно обрабатывать файлы во вложенных папках, отмените параметр *Обрабатывать вложенные папки*. Атрибут *Сжатый* может быть изменен только на дисках с файловой системой NTFS.

Поддерживаются три различных временных атрибута файла:

- время последней модификации;
- время создания файла;
- время последнего доступа.

Для дисков с файловой системой FAT часы, минуты и секунды времени последнего доступа всегда равны нулю.

Если вы не хотите изменять время файла, оставьте соответствующее поле пустым.

Кнопка Текущее позволяет заполнить поля времени файла текущим временем.

Описания файлов. Описания могут быть использованы для того, чтобы связать с файлом текстовую информацию. Описания файлов текущей папки хранятся в этой папке в специальном файле — списке описаний. В нем в начале каждой строки содержится имя описываемого файла и отделенный от него пробелами текст описания.

Описания можно посмотреть в соответствующих режимах просмотра панели файлов. По умолчанию этими режимами являются Описания и Длинные описания.

Команда Описание файлов (<Ctrl+Z>) из Меню файлов предназначена для добавления описаний к выбранным файлам.

Имена списков описаний могут быть изменены в диалоге Описания файлов из меню Меню параметров. В этом диалоге также можно установить режим обновления локальных описаний. Обновление может быть запрещено совсем, разрешено, только если текущий режим просмотра файловой панели показывает описания, или разрешено всегда. По умолчанию FAR устанавливает атрибут Hidden на созданные списки описаний, но вы можете это запретить, включив опцию Устанавливать атрибут Hidden на новые списки описаний в этом же диалоге. Также здесь вы можете указать позицию для выравнивания новых описаний в списке описаний.

Если это разрешено в конфигурации, FAR обновляет описания файлов при копировании, переносе и удалении файлов. Но если команда обрабатывает и часть файлов во вложенных папках, то для этих файлов описания не обновляются.

Меню команд

Предусматривает следующее (см. табл. П1.4—П1.6):

- поиск файла — поиск в дереве папок файлов, удовлетворяющих заданной маске;
- история команд — показать предыдущие команды;
- видеорежим — выбрать количество строк на экране;

- поиск папки — поиск папки в дереве папок;
- история просмотра — показать историю просмотра и редактирования файлов;
- история папок — показать историю смены папок. Элементы истории просмотра и истории смены папок после выбора передвигаются в конец списка. Необходимо использовать <Shift+Enter>, чтобы выбрать элемент без смены его позиции;
- поменять панели — поменять левую и правую панели местами;
- панели вкл/выкл — показать/спрятать обе панели;
- сравнение папок — сравнить содержимое папок;
- меню пользователя — позволяет редактировать главное или местное меню пользователя. Для вставки пункта используется <Ins>, для удаления — <Del>, для редактирования — <F4>;
- ассоциации файлов — показывает список ассоциаций файлов. для вставки новой ассоциации может использоваться <Ins>, для удаления — <Del>, для редактирования — <F4>;
- ссылки на папки — показывает текущие ссылки на папки;
- группы сортировки — позволяет редактировать задаваемые пользователем группы сортировки;
- фильтр панели — позволяет управлять содержимым панели файлов;
- список экранов — показывает список открытых экранов;
- список задач — показывает список активных задач.

Поиск файла (рис. 4.12). Эта команда предназначена для поиска одного или нескольких файлов и папок в дереве папок в соответствии с одной или несколькими разделенными запятыми масками. Также она может быть использована с файловыми системами, поддерживаемыми с помощью внешних модулей (Plugins). Дополнительно может быть указан текст, который должен содержаться в разыскиваемых файлах. В этом случае параметр Учитывать регистр может быть использован для проведения поиска текста с учетом регистра. С помощью кнопки Таблица можно изменить таблицу символов, используемую для поиска текста. Параметр Использовать все таблицы символов заставляет FAR использовать все доступные ему таблицы для поиска текста в файлах с различной кодировкой. Для поиска файлов и в архивах нужно установить опцию Искать в архи-

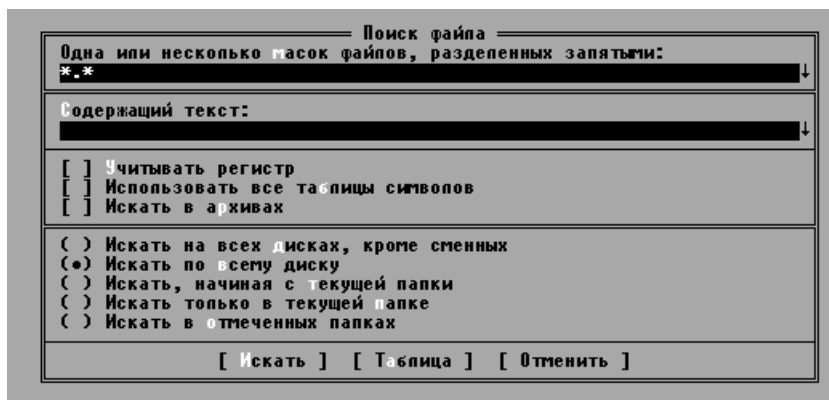


Рис. 4.12. Окно команды Поиск файла

вах. В то же время она существенно замедляет выполнение операции и не позволяет выполнять поиск во вложенных архивах.

Поиск может выполняться на всех дисках, кроме сменных, во всех папках, начиная с корневой, с текущей папки, только в текущей папке или в отмеченных папках. Область поиска сохраняется в конфигурации.

Во время или после завершения поиска можно использовать клавиши управления курсором для передвижения по списку файлов и кнопки для выполнения требуемых действий.

Во время или после завершения поиска доступны следующие кнопки (рис. 4.13):

- **новый поиск** — начать новую операцию поиска;
- **перейти** — прервать поиск, сменить текущую папку и поместить курсор на выбранный файл;
- **смотреть** — просмотр выбранного файла, если поиск не завершен, он будет возобновлен по окончании просмотра;
- **панель** — создать временную панель и заполнить ее найденными файлами;
- **стоп** — прервать поиск (доступна во время поиска);
- **отмена** — закрыть диалог поиска.

Для просмотра и редактирования найденных файлов могут быть использованы <F3> и <F4>, но редактирование и просмотр не поддерживаются для файловых систем подключаемых модулей.

Поиск папки. Эта команда предназначена для быстрого поиска нужной папки в дереве папок. Для выбора папки можно

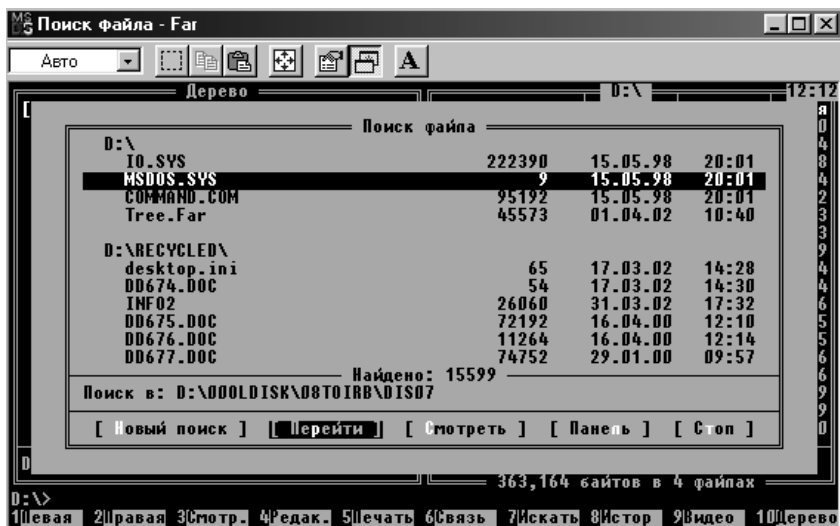


Рис. 4.13. Экран в процессе поиска файлов

использовать клавиши управления курсором или набрать несколько начальных символов имени папки. Нажать <Enter> для перехода в выбранную папку. <Ctrl+R> и <F2> позволяют перечитать дерево папок.

История команд. История команд показывает список выполненных ранее команд. Необходимо выбрать команду и нажать <Enter>, чтобы выполнить ее еще раз; <Shift+Enter>, чтобы выполнить ее в отдельном окне или <Ctrl+Enter>, чтобы поместить ее в командную строку для редактирования. Для выбора команды, кроме клавиш управления курсором и <Enter>, можно использовать подсвеченные буквы. <Del> очищает историю команд. Кроме того, для перехода к предыдущей или следующей команде непосредственно из командной строки можно использовать клавиши <Ctrl+E> или <Ctrl+X> соответственно.

Для сохранения истории команд перед выходом необходимо использовать соответствующую опцию из диалога системных параметров.

Список задач. Список задач показывает активные на данный момент задачи. Каждая строка списка содержит заголовок окна задачи. Из списка задач можно переключиться на окно задачи или удалить задачу, используя клавишу <Del>. Эта операция выполняется немедленно и любая несохраненная информа-

ция данной задачи будет потеряна. Поэтому удаление задач должно использоваться только при необходимости, например, если программа перестала отвечать на запросы пользователя. Список задач может быть вызван либо из Меню команд, либо используя <Ctrl+W>. В последнем случае список задач также может быть вызван из программы просмотра или редактора.

Сравнение папок. Команду сравнения папок можно выполнять, только когда обе панели на экране являются панелями файлов. Она сравнивает содержимое отображаемых в этих панелях папок. Файлы, которые присутствуют только в одной панели, или файлы, чья дата модификации более свежая, чем у файлов с тем же именем в другой панели, становятся помеченными. Вложенные папки не сравниваются. Для сравнения файлов используется их имя, дата и время, но не содержимое.

Меню пользователя. Меню пользователя предназначено для упрощения выполнения часто используемых операций. Оно содержит заданные пользователем команды и последовательности команд, которые могут быть выполнены с использованием этого меню. Меню пользователя может включать вложенные меню. Специальные символы, используемые при Ассоциации файлов, поддерживаются и в командах, и в заголовках команд меню. Составной символ `!<title>?<init>!` может быть использован для ввода дополнительных параметров непосредственно перед выполнением команды. Для редактирования или создания главного или местного меню пользователя применяется команда Меню пользователя из Меню команд. Главное Меню пользователя может быть только одно. Главное меню вызывается в том случае, если для текущей папки отсутствует местное меню. Местное меню может быть расположено в любой папке.

Для выполнения команды из Меню пользователя нужно выбрать ее с помощью клавиш управления курсором и нажать <Enter>. Также можно использовать назначенную для данного пункта меню горячую клавишу.

Можно удалить вложенное меню или пункт меню, используя клавишу <Del>, вставить новое вложенное меню или пункт меню с помощью <Ins> и редактировать существующее вложенное меню или пункт меню с помощью <F4>, <Alt+F4> — для редактирования меню в виде текстового файла. В качестве горячих клавиш для обращения к пунктам меню могут использоваться цифры, буквы и функциональные клавиши (<F1>—<F12>). Если использованы <F1> и <F4>, их первоначальные функции

теряются. В этом случае для редактирования меню может применяться <Shift+F4>.

При редактировании или создании пункта меню нужно ввести горячую клавишу для быстрого доступа к этому пункту, заголовок пункта, который будет отображаться в меню, и последовательность команд для выполнения в случае выбора данного пункта меню.

При редактировании или создании вложенного меню достаточно ввести горячую клавишу и заголовок вложенного меню. Местные меню хранятся в текстовых файлах FarMenu.Ini.

Главное меню по умолчанию хранится в Реестре, но его также можно держать в файле. Если создать местное меню в папке FAR, то оно будет использовано вместо хранящегося в Реестре главного меню.

Ассоциации файлов. FAR позволяет задать три команды, ассоциированные с определенным типом файла, заданным маской:

- команда запуска — выполняется при нажатии <Enter>;
- команда просмотра — выполняется при нажатии <F3>;
- команда редактирования — выполняется при нажатии <F4>.

Ассоциацию можно описать в поле Описание ассоциации.

Новые ассоциации добавляются с помощью команды Ассоциации файлов в Меню команд.

В ассоциированных командах могут использоваться специальные символы (табл. 4.2).

Допускается использование нескольких таких символов в одной строке, например:

```
grep !?Search for:?! !?In:?*.?!|c:\far\far.exe -v -
```

Префикс !<<, указанный перед символом ассоциации файла, заставляет его ссылаться на пассивную панель. Например, !<<! .! обозначает имя текущего файла на пассивной панели.

Примечания:

1. Если для данного файла отсутствуют ассоциированные команды запуска и установлен параметр Использовать стандартные типы в Системных параметрах, то FAR пытается использовать ассоциации Windows для запуска этого типа файлов.

2. Вы можно задать несколько ассоциаций для одного типа файлов и выбирать желаемую ассоциацию из меню.

Используя пункт Применить команду из Меню файлов, можно применить команду к каждому помеченному файлу. Для

Таблица 4.2. Использование специальных символов

Символы	Интерпретация
!!	Символ «!»
!	Длинное имя файла без расширения
!~	Короткое имя файла без расширения
!.!	Длинное имя файла с расширением
!-!	Короткое имя файла с расширением
!+!	Аналогично !-!, но если длинное имя файла утеряно после выполнения команды, FAR восстановит его
!@!	Имя файла, содержащего имена помеченных файлов
!\$!	Имя файла, содержащего короткие имена помеченных файлов
!:	Текущий диск
!\	Текущий путь
! /	Короткое имя текущего пути
!<title>?<init>!	При выполнении команды этот символ заменяется данными, введенными пользователем. <title> и <init> — заголовок и исходный текст строки редактирования

обозначения имени файла должны использоваться те же символы, что и в Ассоциациях файлов.

Например, 'type !.!' будет выводить на экран все помеченные файлы по очереди.

Ссылки на папки. Ссылки на папки позволяют обеспечить быстрый доступ к часто используемым папкам. Для создания ссылки на текущую папку нужно нажать <Ctrl+Shift+N>, где N — '0'..'9'. После этого, чтобы перейти в папку, записанную в ссылке, достаточно нажать <ПравыйCtrl+N>. Если <ПравыйCtrl+N> нажат в строке редактирования, то пункт ссылки будет вставлен в эту строку.

Пункт Ссылки на папки в Меню команд позволяет просматривать, устанавливать, редактировать и удалять ссылки на папки.

Группы сортировки. Группы сортировки могут применяться в панели файлов совместно с сортировкой *по имени или по расширению*. Они активизируются при нажатии <Shift+F11> и позволяют задать правила сортировки файлов, дополняющие уже действующие.

Каждая группа сортировки состоит из одной или нескольких разделенных запятыми масок файлов. Если позиция одной группы сортировки в общем списке групп выше, чем у другой группы,

то при сортировке по возрастанию все принадлежащие к этой группе файлы будут выше, чем принадлежащие к другой группе.

Команда Группы сортировки из Меню команд позволяет удалять, создавать и редактировать группы сортировки, используя клавиши <Del>, <Ins> и <F4>. Группы, находящиеся выше разделителя меню, относятся к началу файловой панели, и все файлы, попавшие в эти группы, будут расположены выше не попавших в них файлов. Группы, находящиеся ниже разделителя меню, относятся к концу файловой панели, и все файлы, попавшие в эти группы, будут расположены ниже не попавших в них файлов.

Фильтр файловой панели. С помощью фильтра можно определить набор типов файлов, которые будут показываться в панели файлов. Меню фильтра состоит из двух частей. В верхней части расположены пользовательские фильтры. С помощью клавиш <Ins>, <Del> и <F4> можно добавлять, удалять и редактировать их. Каждый пользовательский фильтр включает необязательный заголовок и маску файлов либо несколько масок файлов, разделенных запятыми. В нижней части меню фильтра находятся маски всех файлов, содержащихся в данный момент в активной панели файлов.

Для выбора элементов меню фильтров могут использоваться клавиши <Space>, <+> и <->. Элементы, выбранные с помощью пробела или <+>, помечаются символом «+». Если такие элементы присутствуют, то будут показаны только удовлетворяющие им файлы. Элементы, выбранные с помощью <->, помечаются символом «-», и все удовлетворяющие им файлы будут исключены из панели файлов.

Пометка пользовательских фильтров сохраняется в конфигурации.

Когда фильтр используется в панели, это показывается символом <*> после буквы режима сортировки в верхнем левом углу панели.

Прочие инструментальные возможности

Переключение между экранами. FAR позволяет открыть несколько копий встроенной программы просмотра и редактора. Следует использовать <Ctrl+Tab>, <Ctrl+Shift+Tab> или <F12> для переключения между панелями и экранами с этими копиями. <Ctrl+Tab> переключает на следующий экран,

<Ctrl+Shift+Tab> — на предыдущий, <F12> выводит список всех доступных экранов.

Количество фоновых экранов редактирования и просмотра отображается в верхнем левом углу левой панели. Можно запретить показ количества экранов, используя диалог Настройки панели.

Клавиатурные макрокоманды. Клавиатурные макрокоманды могут быть использованы для переопределения стандартных клавиш или комбинаций клавиш FAR или для создания новых клавиатурных команд. Для задания макрокоманды необходимо нажать <Ctrl+.> (Ctrl и клавишу с точкой), желаемую последовательность клавиш, опять <Ctrl+.> и клавишу либо комбинацию клавиш, на которую будет назначена эта макрокоманда. Во время записи макрокоманды в верхнем левом углу экрана выводится символ 'R'.

Чтобы удалить макрокоманду и вернуть клавише ее первоначальную функцию, следует дважды нажать <Ctrl+.> и затем клавишу, на которую назначена макрокоманда.

Дополнительно к стандартным комбинациям клавиш FAR можно назначать макрокоманды на <Ctrl+Shift+буква>, <Ctrl+Alt+буква> и <Alt+Shift+буква>.

FAR поддерживает несколько независимых наборов макрокоманд: макрокоманды оболочки, программы просмотра, редактора, а также некоторые другие типы макрокоманд. Наборы макрокоманд сохраняются по команде Сохранить параметры из Меню параметров.

Для задания дополнительных параметров макрокоманды необходимо начать или завершить ее запись с помощью <Ctrl+Shift+.> вместо <Ctrl+.> и выбрать желаемые опции в появившемся диалоге.

Встроенная программа просмотра

Возможности программы просмотра достаточно очевидны из списка команд, приведенных в табл. 4.3.

Настройки программы просмотра. В этом диалоге можно изменить параметры программы просмотра.

А. Внешняя программа просмотра:

- запускать внешнюю программу просмотра по <F3>;
- запускать внешнюю программу просмотра по <Alt+F3>;

Таблица 4.3. Основные команды программы просмотра

Клавиши	Функция	Клавиши	Функция
<←>	Символ влево	<F7>	Поиск
<→>	Символ вправо	<Shift+F7>, <Space>	Искать дальше
<↑>	Строку вверх	<F8>	Переключить режим просмотра текста DOS/Windows
<↓>	Строку вниз	<Shift-F8>	Выбор пользовательской таблицы символов
<Ctrl+←>	20 символов влево	<Alt-F8>	Изменить текущую позицию
<Ctrl+→>	20 символов вправо	<Num5>, <F3>, <F10>, <Esc>	Выход
<PgUp>	Страницу вверх	<F11>	Вызвать меню «Команды внешних модулей»
<PgDn>	Страницу вниз	<+>	Перейти к следующему файлу
<Home>	В начало файла	<->	Перейти к предыдущему файлу
<End>	В конец файла	<Ctrl+O>	Показать пользовательский экран
<F1>	Помощь	<Alt+BS>, <Ctrl+Z>	Возврат к предыдущей позиции
<F2>	Переключить свертку строк	<ПравыйCtrl+N>	Сохранить текущую позицию (N должно быть равно «0»..«9»)
<F4>	Переключение между текстовым и шестнадцатеричным режимом	<ЛевыйCtrl+N>	Восстановить сохраненную позицию
<F6>	Переключиться в редактор		

Примечания:

1. Пользовательские таблицы символов находятся в папке Addons\Tables папки FAR в виде.reg файлов. Перед использованием любой из таблиц ее нужно установить; для этого нажать <Shift+Enter> на соответствующем файле.

2. Для вызова диалога поиска достаточно просто начать вводить предназначенный для поиска текст.

- команда просмотра — команда для запуска внешней программы просмотра, для указания имени просматриваемого файла использовать специальные символы, описанные в ассоциациях файлов.

Если внешняя программа просмотра назначена на клавишу <F3>, она будет запускаться только в том случае, если ассоции-

рованная программа просмотра для данного типа файлов отсутствует.

Б. Встроенная программа просмотра:

- сохранять позицию файла — сохранять и восстанавливать позицию в недавно просмотренных файлах. Эта опция также вызывает сохранение таблицы символов, использованной при просмотре файла, в случае если эта таблица была установлена пользователем вручную;
- автоопределение таблицы символов — если приходится пользоваться несколькими таблицами символов и установлена таблица с распределением частот символов для выбранного языка, можно включить эту опцию для автоопределения таблицы просматриваемого файла. Очевидно, что корректное определение не гарантируется, особенно для маленьких или нетипичных текстовых файлов. Некоторые таблицы символов помещены в папке Addons\Tables дистрибутива FAR;
- размер табуляции — количество пробелов при показе символа табуляции.

Встроенный редактор

Команды управления встроенным редактором приведены в табл. 4.4 и 4.5.

Таблица 4.4. Команды управления курсором и удаления

Клавиши	Функция
Команды управления курсором	
<←>	Символ влево
<→>	Символ вправо
<↑>	Строку вверх
<↓>	Строку вниз
<Ctrl+←>	Слово влево
<Ctrl+→>	Слово вправо
<Ctrl+↑>	Прокрутка экрана вверх
<Ctrl+↓>	Прокрутка экрана вниз

Окончание табл. 4.4

Клавиши	Функция
<PgUp>	Страницу вверх
<PgDn>	Страницу вниз
<Home>	В начало строки
<End>	В конец строки
<Ctrl+Home>	В начало файла
<Ctrl+End>	В конец файла
<Ctrl+N>	В начало экрана
<Ctrl+E>	В конец экрана
Команды удаления	
<Del>	Удалить символ (также может быть удален блок, в зависимости от настроек редактора)
<BS>	Удалить символ слева
<Ctrl+Y>	Удалить строку
<Ctrl+K>, <Alt+D>	Удалить до конца строки
<Ctrl+BS>	Удалить слово слева
<Ctrl+T>, <Ctrl+Del>	Удалить слово справа

Таблица 4.5. Команды работы с блоками и прочие команды

Команда	Функция
Операции над блоками	
<Shift+Клавиши курсора>	Пометка блока
<Ctrl+Shift+клавиши курсора>	Пометка блока
<Alt+серые клавиши курсора>	Пометка вертикального блока
<Alt+Shift+клавиши курсора>	Пометка вертикального блока
<Ctrl+A>	Пометить весь текст
<Ctrl+U>	Снять пометку с блока
<Shift+Ins>, <Ctrl+V>	Скопировать блок из Буфера Обмена

Окончание табл. 4.5

Команда	Функция
<Shift+Del>, <Ctrl+X>	Переместить блок в Буфер Обмена
<Ctrl+Ins>, <Ctrl+C>	Скопировать блок в Буфер Обмена
<Ctrl+Num+>	Добавить блок в Буфер Обмена
<Ctrl+D>	Удалить блок
<Ctrl+P>	Копировать блок в текущую позицию курсора (только в режиме постоянных блоков)
<Ctrl+M>	Перенести блок в текущую позицию курсора (только в режиме постоянных блоков)
<Alt+U>	Сдвинуть блок влево
<Alt+I>	Сдвинуть блок вправо
Прочие операции	
<F1>	Помощь
<F2>	Сохранить файл
<Shift+F2>	Сохранить файл как Name
<F6>	Переключиться в программу просмотра
<F7>	Поиск
<Ctrl+F7>	Замена
<Shift+F7>	Продолжить поиск
<F8>	Переключение между DOS/Windows текстом
<Shift+F8>	Выбор пользовательской таблицы символов
<Alt+F8>	Перейти на указанную строку
<F10>, <Esc>	Выход
<Shift+F10>	Сохранение и выход
<F11>	Вызвать меню Команды внешних модулей
<Alt+BS>, <Ctrl+Z>	Отмена действия
<Ctrl+L>	Запретить модификацию редактируемого текста
<Ctrl+O>	Показать пользовательский экран
<Ctrl+Q>	Рассматривать следующую комбинацию клавиш как код символа
<ПравыйCtrl+N>	Сохранить текущую позицию (N должно быть равно '0'...'9')
<ЛевыйCtrl+N>	Восстановить сохраненную позицию

Настройки редактора. В этом диалоге можно изменить используемые по умолчанию параметры внешнего и встроенного редакторов.

А. Внешний редактор:

- запускать внешний редактор по <F4>;
- запускать внешний редактор по <Alt+F4>;
- команда редактирования — команда для запуска внешнего редактора. Для указания имени просматриваемого файла использовать специальные символы, описанные в Ассоциациях файлов.

Если внешний редактор назначен на клавишу <F4>, он будет запускаться только в том случае, если ассоциированный редактор для данного типа файлов отсутствует.

Б. Встроенный редактор:

- пробелы вместо табуляции — заменять символ <Tab> на соответствующее количество пробелов;
- постоянные блоки — не удалять пометку блока при перемещении курсора;
- <Del> удаляет блоки, если есть помеченный блок, <Del> будет удалять этот блок, а не символ под курсором;
- автоотступ — включает режим автоотступа при вводе текста;
- сохранять позицию файла — сохранять и восстанавливать позицию в недавно отредактированных файлах. Эта опция также вызывает сохранение таблицы символов, использованной при редактировании файла, в случае если эта таблица была установлена пользователем вручную;
- курсор за пределами строки — разрешить перемещение курсора за пределы строки;
- автоопределение символов, размер табуляции — то же, как и для программ просмотра.

Помощь

Страницы помощи могут иметь ссылки, ведущие к другим страницам. Для перемещения между ссылками вы можете использовать <Tab> и <Shift+Tab>. Нажатие <Enter> вызовет страницу, соответствующую выбранной ссылке. Этот же результат можно получить с помощью мыши, нажав ее кнопку на требуемой ссылке.

Если текст помещается в окне помощи не полностью, то появляется полоса прокрутки. В этом случае текст можно перемещать вверх и вниз с помощью клавиш управления курсором.

Нажатие <Alt+F1> или <BS> приводит к переходу на предыдущую страницу. <Shift+F1> вызывает содержание помощи. Чтобы получить помощь по внешним модулям, необходимо нажать <Shift+F2>. Для переключения между полноэкранным режимом отображения помощи и выводом текста в окне можно использовать <F5>.

4.5. Программная оболочка Dosshell

Программа Dosshell представляет собой текстовую оболочку, предназначенную для работы с DOS [17]. В оболочке Dosshell можно запускать прикладные программы на исполнение, осуществлять операции с файлами и дисками. Возможности Dosshell шире, чем PCTools, а ряд операций выполняет только Dosshell. При копировании файлов используется технология Drag and Drop («переместить и отпустить»).

Исходный кадр оболочки MS DOS Shell

После вызова оболочки MS DOS Shell на экране появляется информационный кадр, показанный на рис. 4.14.

Начальный информационный кадр включает шесть областей (окон):

- строку основного меню;
- линейку дисков;
- изображение дерева каталогов выбранного диска;
- список файлов выбранного каталога;
- список программ (программных групп и программных элементов);
- список активных задач.

В любой момент одно из окон является активным и выделяется подсветкой заголовка. Смена активного окна осуществляется последовательным нажатием клавиши <Tab> или щелчком левой клавиши мыши на заголовке требуемого окна. После загрузки оболочки активной становится линейка имен дисков.

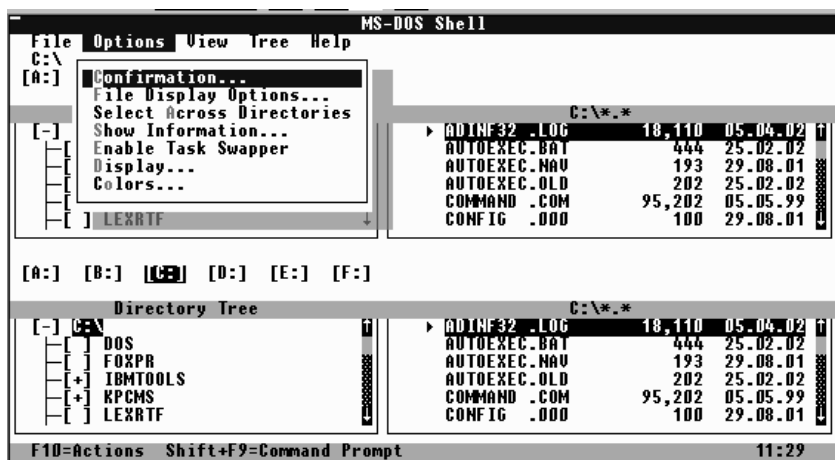


Рис. 4.14. Начальный информационный кадр оболочки MS DOS Shell

Линейка меню в верхней части экрана File Options View Tree Help служит для задания режимов работы и активизации опций оболочки. Активизация требуемого пункта меню осуществляется следующим образом:

- активизацией всей линейки меню клавишами <Alt> или <F10>, переводом курсора с помощью клавиш <←>, <→> на требуемый пункт и нажатием клавиши <Enter>;
- активизацией всей линейки меню клавишами <Alt> или <F10> и нажатием клавиши с выделенной буквой требуемого пункта меню (F для File, O для Options, V для View, T для Tree и H для Help);
- щелчком левой клавиши мыши на требуемом пункте меню.

Линейка имен дисков [A:] [B:] [C:] [D:] позволяет сменить диск, информация из которого выводится на экран. Смена диска осуществляется переводом курсора с помощью клавиш <←>, <→> на обозначение требуемого диска и нажатием клавиши <Enter> (при активной линейке дисков) или щелчком левой клавиши мыши на обозначении требуемого диска (при любом активном окне).

Левая верхняя четверть информационного кадра служит для изображения дерева каталогов выбранного диска, а правая верхняя — для вывода списка файлов выбранного каталога.

Смена выбранного каталога осуществляется клавишами <↑>, <↓> или щелчком левой клавиши мыши на требуемом каталоге,

вывод на экран ветви, нижележащей относительно выбранного каталога, — клавишей <+> или щелчком левой клавиши мыши на знаке «+» развертываемого каталога. Свертывание ветви каталога осуществляется нажатием клавиши <->, когда выбранным является свертываемый каталог, или щелчком левой клавиши мыши на знаке «-» свертываемого каталога (при любом выбранном каталоге).

Операции с файлами

Любая операция с файлом требует, чтобы этот файл был выделен. Групповые операции требуют выделения группы файлов. Выделение файла осуществляется активизацией окна со списком файлов с помощью клавиши <Tab> и переводом курсора с помощью клавиш <↑>, <↓> на выбираемый файл. Щелчок левой клавиши мыши на требуемом файле активизирует окно файлов и выделяет файл.

Для выделения всех файлов выбранного каталога можно воспользоваться пунктом меню File\Select All или нажать <Ctrl+/>.

Для выделения группы последовательных файлов следует при нажатой клавише <Shift> нажимать клавиши <↑> или <↓>. Каждое нажатие будет добавлять очередной файл к группе выделенных. Для выделения группы последовательных файлов с помощью мыши следует щелкнуть левой клавишей мыши на первом файле и при нажатой клавише <Shift> щелкнуть левой клавишей мыши на последнем из выделяемых файлов.

Для выделения группы произвольных (не последовательных в списке) файлов с помощью клавиатуры следует, выделив первый файл, перейти в режим добавления нажатием <Shift+F8> (повторное нажатие <Shift+F8> выключает этот режим). Далее, устанавливая курсор с помощью клавиш <↑>, <↓> на требуемые файлы, включать их в группу выделенных клавишей пробела (повторное нажатие клавиши пробела отменяет выделение данного файла).

Для выделения группы произвольных файлов с помощью мыши следует при нажатой клавише <Ctrl> щелкать левой клавишей мыши на именах требуемых файлов (отмена выделения выполняется повторным щелчком левой клавиши мыши на имени файла при нажатой клавише <Ctrl>).

Для того чтобы создать группу выделенных файлов из разных каталогов, следует включить экранную клавишу Options\Select Across Directories. При повторной активизации меню Options в начале строки Select Across Directories будет стоять метка, свидетельствующая о включенном состоянии режима.

Оболочка MS DOS Shell предоставляет средства поиска файлов по имени или по шаблону групповой операции как в пределах выбранного каталога, так и на всем выбранном диске. Поиск файлов активизируется с помощью пункта меню File\Search. На экран выводится диалоговая рамка, в которой можно указать имя искомого файла или шаблон групповой операции (по умолчанию действует шаблон *.*), а также включить или выключить поиск по всему диску экранной клавишей Search entire disk.

Копирование выделенных файлов осуществляется с помощью пункта меню File\Copy. На экран выводится диалоговая рамка, в которой следует указать спецификацию файла-приемника.

Перемещение выделенных файлов (т. е. копирование с одновременным удалением файлов-источников) осуществляется с помощью пункта меню File\Move.

Удаление выделенных файлов осуществляется с помощью пункта меню File\Delete или нажатием клавиши <Del>. Если включена экранная клавиша Option\Confirmation\Confirm on Delete, на экран выводится предупреждающее сообщение с требованием подтверждения операции. Если экранная клавиша Option\Confirmation\Confirm on Delete выключена, предупреждающее сообщение не выводится.

Пункт меню File\Rename позволяет переименовывать выделенные файлы (рис. 4.15). Если выделена группа файлов, диалоговая рамка переименования, куда следует вписать новое имя файла, выводится отдельно для каждого переименовываемого файла.

Содержимое любого (не только текстового) файла можно просмотреть на экране с помощью пункта меню View\File Contents или посредством нажатия клавиши <F9>. Текстовые файлы выводятся в текстовом режиме, двоичные файлы — в шестнадцатеричном, однако и те и другие файлы можно наблюдать в обоих режимах. Переключение режимов осуществляется той же клавишей <F9>. Прокрутка содержимого длинных файлов выполняется клавишами <↑>, <↓>, а постраничная прокрутка —

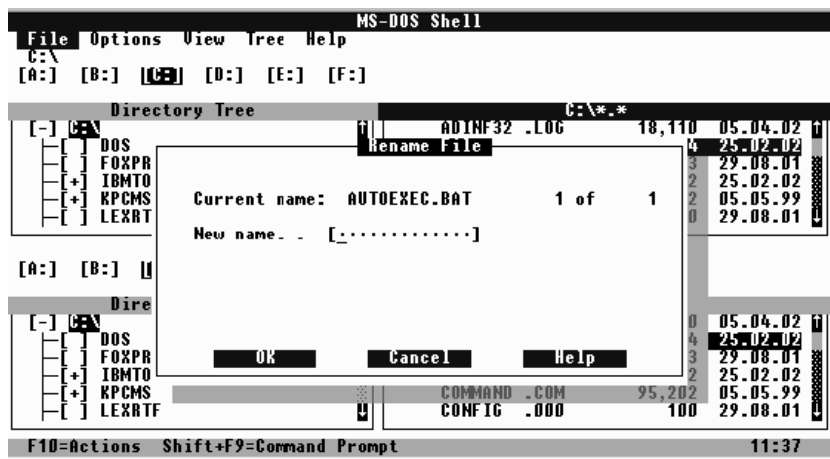


Рис. 4.15. Переименование файла

клавишами <PgUp>, <PgDn>. Для выхода из режима просмотра достаточно нажать клавишу <Esc>.

Оболочка MS DOS Shell позволяет изменять атрибуты r (только для чтения), s (системный), h (скрытый) и a (архивный) выделенных файлов.

Если до запуска оболочки MS DOS Shell загружена резидентная программа MS DOS PRINT, из оболочки можно выполнять печать файлов на принтере. При этом должно быть активным окно со списком файлов. В нем следует выбрать файлы, подлежащие выводу на печать, после чего активизировать пункт меню File\Print.

Любое расширение имени файла может быть связано с некоторой программой, так чтобы открытие конкретного файла с таким расширением автоматически запускало связанную программу для обработки именно этого файла. Например, файлы с расширениями txt или doc можно связать с текстовыми редакторами, а файлы с расширениями zip или arj — с соответствующими архиваторами. Связь расширений с программами осуществляется с помощью пункта меню File\Associate. После установления связи нажатие клавиши <Enter> или двойной щелчок левой клавиши мыши на имени файла запускает связанную программу для этого файла.

В состав информационного кадра оболочки MS DOS Shell входит изображение дерева каталогов выбранного диска и спи-

ска файлов выбранного каталога этого диска. Формат информационного кадра можно изменять с помощью пункта View основного меню.

Пункт меню View\Dual File Lists позволяет вывести на экран одновременно два дерева каталогов со списками файлов. Каталоги могут принадлежать одному или разным дискам.

Пункт меню View\All Files позволяет вывести на экран список всех файлов выбранного диска, включая системные файлы. Работая в таком формате, можно с помощью линейки дисков выполнять смену выбранного диска.

Перейти из режима списка всех файлов в режим изображения каталогов и файлов можно с помощью пункта меню View\Single File List (дерево каталогов и список файлов занимают весь экран). В режим списка программ переход осуществляется с помощью пункта View\Program List (список программных групп и программных элементов занимает весь экран), в режим двух каталогов — с помощью пункта View\Dual File Lists, а в исходный режим всех четырех окон — с помощью пункта View\Program\File Lists.

Поскольку оболочка MS DOS Shell хранит информацию об информационном кадре в памяти, изменение содержимого диска (например, создание новых файлов) может не найти отражения в информационном кадре. Для обновления информационного кадра после модификации содержимого диска следует воспользоваться пунктом View\Refresh или клавишей <F5>.

Операции с деревом каталогов

Для смены диска, информация из которого выводится в информационный кадр с помощью клавиатуры, можно при нажатой клавише <Ctrl> нажать букву, соответствующую обозначению диска. В режиме двух каталогов смена диска осуществляется в выбранной половине информационного кадра. Другой способ смены диска — с помощью клавиши <Tab> выбрать линейку дисков, затем клавишами <←>, <→> выбрать требуемый диск и нажать <Enter>.

Для смены диска с помощью мыши достаточно щелкнуть левой клавишей мыши на обозначении требуемого диска. Двойной щелчок левой клавиши мыши приводит к смене диска с одновременным обновлением информационного кадра.

Для вывода на экран дополнительной информации о выбранных диске, каталоге и файле следует активизировать пункт меню `Options\Show Information`. Выход из этого кадра осуществляется выбором экранной клавиши `Close` или нажатием `<Esc>`. Вывод дополнительной информации возможен только, если активными являются окна с каталогами или файлами (не с программами).

Если выбран некоторый каталог дерева каталогов, в правое окно выводится список файлов выбранного каталога. Смена выбранного каталога осуществляется клавишами `<↑>`, `<↓>` или щелчком левой клавиши мыши на требуемом каталоге.

Для вывода в информационный кадр ветви, нижележащей относительно выбранного каталога, следует воспользоваться пунктом меню `Tree\Expand One Level` или нажать клавишу `<+>`. Развертывание каталога с помощью мыши выполняется щелчком левой клавиши мыши на знаке `«+»` слева от имени каталога.

Вывод на экран каталогов всех уровней, нижележащих относительно выбранного, осуществляется нажатием клавиши `<*>` или с помощью пункта меню `Tree\Expand Branch`. Вывод на экран всего дерева каталогов выбранного диска осуществляется нажатием клавиш `<Ctrl+*>` или с помощью пункта меню `Tree\Expand All`.

Свертывание нижележащих уровней каталогов относительно выбранного осуществляется нажатием клавиши `<->` или с помощью пункта меню `Tree\Collapse Branch`. Свернуть любую ветвь каталогов можно с помощью мыши, щелкнув левой клавишей на знаке `«-»` слева от свертываемого каталога.

Для создания нового каталога следует выбрать тот каталог, в котором предполагается создать новый, после чего выбрать пункт меню `File\Create Directory`. На экран выводится диалоговая рамка, в которую следует вписать имя создаваемого каталога.

Удаление выбранного каталога выполняется с помощью пункта меню `File\Delete` или просто нажатием клавиши `<Del>`. Если включена экранная клавиша `Option\Confirmation\Confirm on Delete`, на экран выводится предупреждающее сообщение с требованием подтверждения операции. Если экранная клавиша `Option\Confirmation\Confirm on Delete` выключена, предупреждающее сообщение не выводится. Удалить можно только пустой каталог.

С помощью пункта меню `File\Rename` можно переименовать выбранный каталог.

Запуск программ

Запустить программу можно как из списка файлов, так и из списка программ. Для запуска программы из списка файлов с помощью клавиатуры следует выделить программный файл или связанный с ним файл данных и нажать клавишу <Enter> или выбрать пункт меню File\Open. Для запуска программы из списка файлов с помощью мыши следует дважды щелкнуть левой клавишей мыши на имени программного файла или связанного с ним файла данных.

Для запуска программы из списка программ с помощью клавиатуры следует открыть группу, содержащую требуемый программный элемент, выделить этот элемент и нажать клавишу <Enter> или выбрать пункт меню File\Open. Для запуска программы из списка программ с помощью мыши следует дважды щелкнуть левой клавишей мыши на требуемом программном элементе.

Для запуска программы с командной строки DOS из главной группы списка программ следует открыть главную группу Main и активизировать программный элемент Command Prompt. Оболочка загружает вторую копию командного процессора и передает ему управление. Пользователь может вводить с клавиатуры любые командные строки DOS, запуская системные или прикладные программы (в том числе и неоднократно). Для возвращения в оболочку MS DOS Shell следует ввести с клавиатуры команду EXIT.

Для запуска программы, отсутствующей в настоящий момент в информационном кадре, можно воспользоваться пунктом меню File\Run. При активизации этого пункта на экран выводится диалоговая рамка, в которой можно напечатать любую требуемую командную строку DOS.

Запуск программ в многозадачном режиме. При активизации механизма свопинга (выгрузки и загрузки выполняемых программ) возникает возможность одновременной работы с несколькими программами. Каждая программа, запускаемая в этом режиме, добавляется в список активных программ. В процессе выполнения любой активной программы можно выйти на время в оболочку MS DOS Shell или перейти в любую другую активную программу. Описываемая процедура имеет смысл только для программ, работа с которыми осуществляется длительное время в интерактивном режиме диалога с пользователем. К таким программам относятся, например, текстовые редакторы и процессо-

ры, администраторы баз данных и электронных таблиц, системы программирования с полноэкранным интерфейсом и др.

Для включения механизма свопинга следует выбрать пункт меню Options\Enable Task Swapper. Перед обозначением пункта появляется метка, свидетельствующая об активизации многозадачного режима, а в нижней правой части информационного кадра появляется окно списка активных задач (пока пустое).

Имеется два способа запуска программ в многозадачном режиме: с немедленным переходом из оболочки в запускаемую программу и без выхода из оболочки. Последний режим удобен в тех случаях, когда пользователь хочет сначала запустить целую группу программ и лишь затем начать работать с ними в требуемой последовательности.

Запуск программы с немедленным переходом в нее осуществляется обычным образом, из списка файлов или программных элементов. На экран выводится информационный кадр запущенной программы, а ее имя заносится оболочкой в список активных задач.

Запуск программы без выхода из оболочки осуществляется нажатием при выделенном имени программы комбинации клавиш <Shift+Enter> или двойным щелчком мыши на имени программы при нажатой клавише <Shift>. Таким образом можно занести в список активных задач несколько программ, оставаясь в оболочке MS DOS Shell. После этого запуск требуемой программы из списка активных программ осуществляется выбором ее имени и нажатием клавиши <Enter> или двойным щелчком левой клавиши мыши на имени программы.

Возврат из текущей программы в оболочку MS DOS Shell осуществляется нажатием <Ctrl+Esc>.

Завершение работы с оболочкой MS DOS Shell возможно только после завершения всех активных программ их внутренними средствами.

Переключение между программами в многозадачном режиме. Для переключения из активной задачи в оболочку MS DOS Shell следует нажать <Ctrl+Esc>. Для обратного переключения из оболочки MS DOS Shell в активную задачу следует выбрать ее в списке активных задач.

Циклическое переключение программ осуществляется клавишей <Tab> при нажатой клавише <Alt>. При каждом нажатии <Tab> на экране появляется кадр с именем очередной программы. Для ее активизации достаточно отпустить клавишу <Alt>.

Циклическое переключение в обратном направлении осуществляется клавишей <Tab> при нажатых клавишах <Shift> и <Alt>.

При наличии нескольких активных задач нажатие <Alt+Esc> осуществляет переключение на следующую программу из списка, а нажатие <Shift+Alt+Esc> — на предыдущую (в число активных задач включается и оболочка MS DOS Shell). При наличии двух активных задач переключение из одной в другую осуществляется нажатием <Alt+Tab>.

С помощью пункта меню File\New\Add Program или File\Properties программе можно назначить горячие клавиши запуска из числа сочетаний <Ctrl+Буква> <Alt+Буква> или <Shift+Буква>.

Использование интерактивного справочника MS DOS Shell

Оболочка MS DOS Shell содержит подробный интерактивный справочник по всем элементам оболочки и правилам работы с ней. Доступ к справочнику осуществляется с помощью пункта верхнего меню Help. Справочник построен в виде ссылочной базы данных, что дает возможность пользователю быстро найти необходимую ему информацию, а также дополнительные сведения и пояснения (рис. 4.16).

Пункт верхнего меню Help предназначен для более или менее систематического изучения оболочки. При необходимости быстро получить справку по используемому в настоящий момент средству оболочки следует нажать клавишу <F1>. На экран будет выведен раздел справочника по конкретному средству (так называемый контекстный поиск).

В нижней части информационного кадра справочника располагаются пять экранных клавиш. Для активизации экранных клавиш следует нажать клавишу <Tab>, которая последовательно выделяет заголовки тем, входящих в текст справки (если они есть), и затем экранные клавиши.

Назначение клавиш:

Close — закрытие текущего окна справочника и возврат в оболочку MS DOS Shell. Для того же служит клавиша <Esc>;

Back — возврат в предыдущий информационный кадр справочника;

Index — вывод на экран списка тем интерактивного справочника оболочки MS DOS Shell;



Рис. 4.16. Указатель к содержанию MS-DOS Shell Help

Keys — вывод на экран раздела справочника о горячих и управляющих клавишах оболочки MS DOS Shell;

Help — вывод на экран раздела справочника о работе с интерактивным справочником оболочки MS DOS Shell.

4.6. Менеджер файлов и Web-обозреватель Konqueror

Konqueror (компонента программного обеспечения KDE Unix/Linux) является менеджером файлов и позволяет выполнять как простые операции (копирование, перемещение), так и достаточно сложные, такие, как просмотр локальных и удаленных дисков. Konqueror отображает содержимое каталогов различными способами, а также показывает мини-изображения файлов, позволяет легко определять и изменять свойства файлов и каталогов, а также запускать приложения нажатием кнопки мыши.

В качестве Web-обозревателя Konqueror полностью совместим со стандартом HTML 4.01, имеет встроенную поддержку языка JavaScript (ECMA-262), CSS (каскадные таблицы стилей) и двунаправленного письма (арабский и иврит), позволяет запускать приложения Java, поддерживает SSL (исполь-

зующийся для безопасного сообщения в Internet), а также дополнительные модули Netscape (в частности для проигрывания медиафайлов Flash, RealAudio и RealVideo). Среди особых возможностей — автодополнение вводимого текста и адресов Internet, импорт закладок из других обозревателей, открытие нескольких Web-страниц в одном окне, а также выполнение функций клиента FTP.

Konqueror позволяет просматривать рисунки и документы без запуска специальных приложений. Для этого он использует встраиваемые компоненты (KDE Parts) этих приложений: KView — для просмотра рисунков, KDVI — для просмотра файлов в формате DVI, KGhostView — для файлов в формате PostScript, компоненты различных приложений KOffice — для соответствующих типов файлов.

Во внешнем виде Konqueror практически все может быть настроено по вкусу пользователя — от стиля в целом, размера текста и значков до выбора необходимых панелей инструментов, пунктов меню и назначения собственных клавиш быстрого вызова. Различные конфигурации Konqueror можно сохранить для последующего использования. Приводимые ниже иллюстрации относятся к версии Linux SuSE 10.0.

При работе с Konqueror, как и с любым другим приложением KDE, целесообразно использовать трехкнопочную мышь. Однако, если используется двухкнопочная мышь, нажатие средней кнопки можно заменить одновременным нажатием левой и правой (для этого необходимо соответствующим образом настроить систему).

В то время как в системах Windows для открытия документов обычно используется двойной щелчок мышью, следует отметить, что Konqueror, как и другие программы KDE, по умолчанию воспринимает одиночный щелчок.

Запуск Konqueror

Поскольку Konqueror выполняет функцию как менеджера файлов, так и обозревателя Internet, он автоматически переключается в необходимый режим в процессе работы. Запуск может осуществляться следующими способами:

- если щелкнуть по значку  (на рабочем столе или на панели) левой кнопкой, Konqueror запустится в режиме ме-

менеджера файлов, если же щелкнуть по значку , то Konqueror будет запущен в режиме обозревателя Internet;

- если в Главном меню (К) выбрать пункт Internet\ Web-браузер Konqueror, то он будет запущен в режиме обозревателя, а при выборе пункта Домой перейдет в режим менеджера файлов;
- при нажатии <Alt+F2> появится окно Выполнить команду (см. рис. 3.18, а); если набрать в нем konqueror (строчными буквами) и нажать <Enter> или кнопку Выполнить, Konqueror будет запущен в режиме менеджера файлов. Чтобы запустить его в режиме обозревателя, необходимо ввести любой Internet-адрес, например <http://www.konqueror.org>;
- Konqueror автоматически запускается в режиме менеджера файлов, если щелкнуть на рабочем столе по знаку, который представляет собой каталог, диск или Корзину.

Основные элементы интерфейса Konqueror

Рассмотрим основные элементы окна Konqueror (рис. 4.17). На этом рисунке показано, как выглядит главное окно менеджера файлов. В самом верху находится заголовок окна Konqueror, который работает так же, как и заголовки других программ в KDE (рис. 4.17, 1). Если щелкнуть по нему правой кнопкой, можно воспользоваться контекстным меню заголовка окна (рис. 4.17, 2).

Под заголовком находится главное меню (рис. 4.17, 3). Щелкнув мышью по одному из его пунктов, можно вызвать вертикальное меню (его можно вызвать также с помощью комбинации клавиш <Alt+подчеркнутая буква> в названии нужного пункта). Например, нажав <Alt+П>, можно вызвать меню Правка (рис. 4.18).

На панели инструментов (рис. 4.17, 4) находятся кнопки для наиболее распространенных действий. Если в Центре управления в разделе Внешний вид и темы\Стиль включено отображение подсказок, то, наведя мышью на какую-либо из кнопок, можно увидеть возле указателя мыши текст с кратким описанием этой кнопки.

У некоторых кнопок, как, например,  (Вверх) и  (Назад) (см. рис. 4.17), в нижнем правом углу есть небольшой чер-

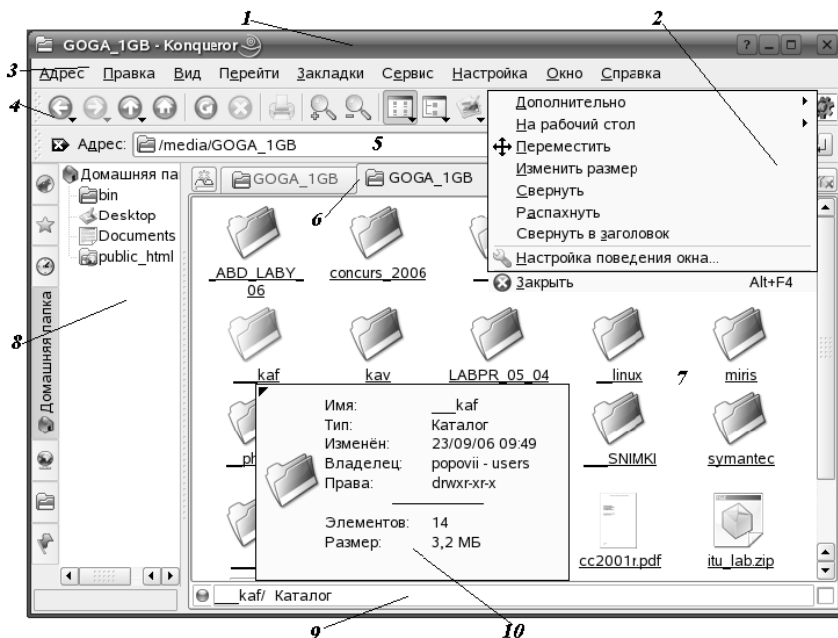


Рис. 4.17. Основные элементы окна Konqueror:

1 — заголовок окна; 2 — меню заголовка; 3 — главное меню; 4 — панель инструментов; 5 — строка адреса; 6 — строка закладок; 7 — главная часть окна; 8 — панель навигации; 9 — строка состояния; 10 — контекстная справка об объекте под курсором

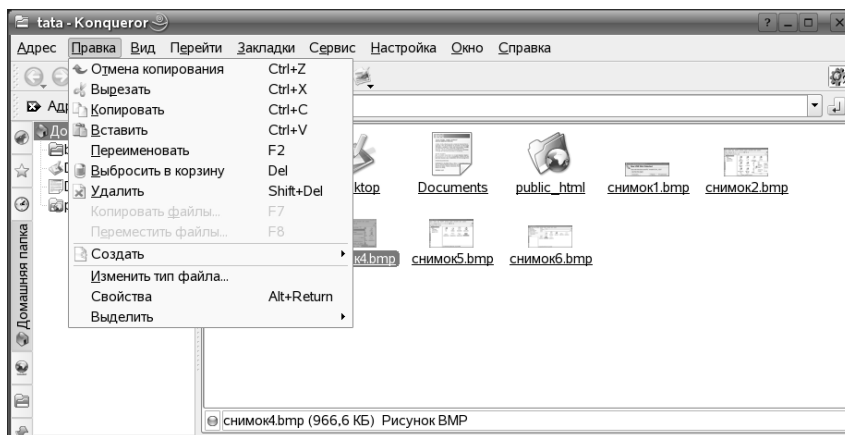


Рис. 4.18. Вертикальное меню Правка

ный треугольник, при нажатии на который появится выпадающее меню.

В строке адреса (рис. 4.17, 5) показывается адрес текущего каталога, страницы в Internet или файла. Чтобы перейти к другому адресу, следует ввести его в это поле и нажать <Enter> на клавиатуре или кнопку **Перейти** справа. С помощью кнопки  можно очистить поле ввода.

Под строкой адреса находится панель закладок (рис. 4.17, 6), где могут размещаться адреса часто посещаемых страниц.

Ниже находятся — главная часть окна Konqueror, в которой отображается каталог, Web-страница, документ или рисунок (рис. 4.17, 7), а также панель навигации (рис. 4.17, 8). С помощью пункта меню **Окно** (рис. 4.19) можно разделить главное окно на два (см. рис. 4.26) или более (что часто бывает удобно для перетаскивания элементов из одного окна в другое) или указать, что все окна должны отображаться во вкладках в одном окне Konqueror.

В нижней части окна интерфейса находится строка состояния (рис. 4.17, 9), в которой отображается общая информация об объекте, на котором находится указатель мыши. Если окно разделено на несколько панелей, у каждой из них будет своя строка состояния (см. рис. 4.26), на которой также размещается индикатор активности панели. Щелчок по строке состояния правой кнопкой мыши позволяет вызвать контекстное меню.

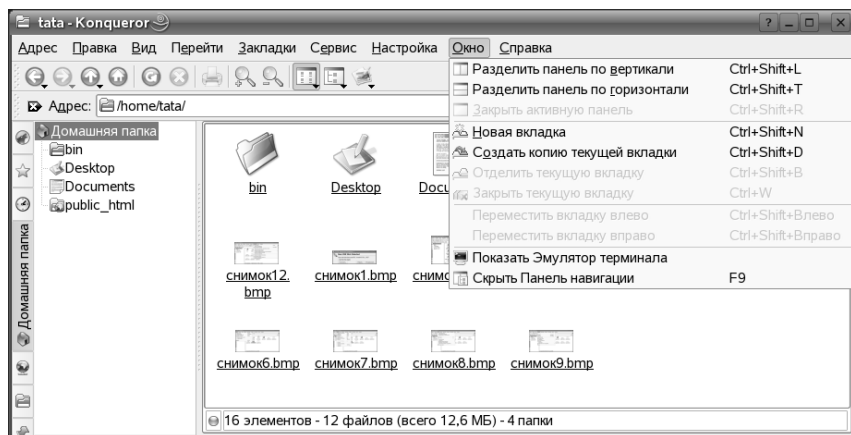


Рис. 4.19. Пункт меню **Окно**

Функции кнопок мыши. Активизация элементов окна Конфигуратора кнопками мыши имеет разновидности, представленные в табл. 4.6. Здесь указано, к каким именно результатам приводит нажатие тех или иных кнопок мыши при размещении курсора на соответствующих элементах экрана.

Таблица 4.6. Функции кнопок мыши

Кнопка мыши	Объект	Результат
Левая	Значок на панели инструментов	Будет выполнено действие, которое присвоено этой кнопке
	Пункт меню	Появится выпадающее вертикальное меню
	Пункт выпадающего меню	Будет выполнено соответствующее действие
	Кнопка на панели закладок	Откроется страница, на которую эта кнопка ссылается
	Ссылка на Web-страницу	Переход к странице, на которую указывает ссылка
	Значок или имя каталога	Переход в этот каталог
	Имя файла или значок	Будет выполнено действие, наиболее подходящее данному типу файлов ¹ . Например, можно просмотреть Web-страницу, текст, рисунок или документ KOffice в окне Конфигуратора (однако правка будет недоступна)
Правая	То же ²	То же ²
	Свободная часть заголовка окна	Вызов специального меню, которое позволяет настроить положение окна Конфигуратора, а также внешний вид программ KDE в целом
	Свободная часть панели инструментов	Вызов специального меню, которое позволяет указать расположение конкретной панели (вверху, внизу, слева или справа), а также установить размер кнопок на этой панели и режим их отображения (только значки, только текст или значки с текстом)
	Строка адреса	Будет показано меню, позволяющее выполнять операции с буфером обмена (такие, как вставка, копирование, очистка), а также изменять параметры автодополнения адресов
	Свободная часть панели закладок	Вызов меню, которое позволяет изменить расположение панели, а также режим отображения кнопок на ней (только значки, только текст или значки с текстом)

Окончание табл. 4.6

Кнопка мыши	Объект	Результат
Правая	Любая свободная часть основного окна	Вызов специального меню, в котором будут, в частности, пункты Вверх, Назад, Вперед и Обновить
	Имя или значок файла	Konqueror выделит этот файл, а также покажет меню, которое позволяет вырезать файл, переместить, скопировать, удалить, добавить в список закладок, открыть его соответствующей программой, переименовать его, а также изменить тип файла или его свойства
	Строка состояния	Можно добавить или удалить панель внутри основного окна Konqueror

¹ Konqueror определяет тип файла по его расширению, а если таким образом тип определить не удастся, то по содержимому. Типы файлов и связанные с ними действия можно изменить, воспользовавшись разделом Привязки файлов в окне настройки Konqueror (Настройка\Настройка Konqueror\...).

² Щелчок средней кнопкой мыши имеет ту же функцию, что и щелчок левой кнопкой, за исключением того, что действие практически всегда происходит в новом окне (или в новой вкладке основного окна, если в разделе Поведение окна настройки Konqueror Настройка\Настроить Konqueror... включен режим Открывать ссылки в новых вкладках, а не окнах). Если щелкая по ссылке, будет удерживаться нажатой клавиша <Shift>, то ссылка будет открыта в новом окне, но в фоновом режиме (текущее окно останется активным). Если щелкнуть средней кнопкой мыши по свободной части окна (не по ссылке или значку файла), Konqueror скопирует содержимое буфера обмена в строку адреса и попытается открыть его как Internet-адрес.

Панель навигации. Панель навигации находится в левой части окна Konqueror (рис. 4.20). Включить ее (или отключить, если она включена) можно с помощью пункта меню Окно\Показать панель навигации или клавишей <F9>.

Данная панель содержит несколько вкладок и чтобы перейти к одной из них, необходимо щелкнуть по ее значку. Чтобы отключить Панель навигации, оставив только значки вкладок, достаточно щелкнуть по значку текущей вкладки. Перечень вкладок приводится в табл. 4.7.

Настроить внешний вид панели навигации можно следующим образом: щелкнуть правой кнопкой мыши по свободному пространству под самой нижней вкладкой, при этом появится меню, предлагающее возможности, представленные в табл. 4.8.

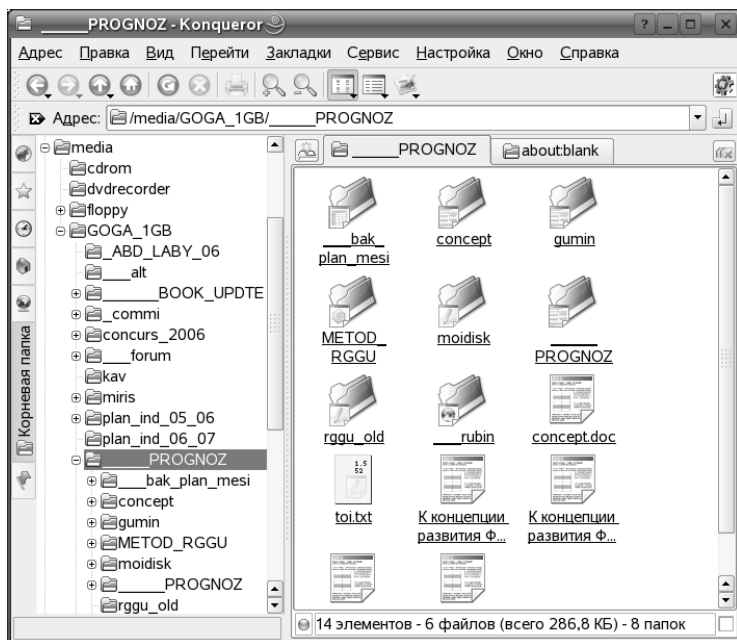


Рис. 4.20. Панель навигации

Таблица 4.7. Вкладки Панели навигации Konqueror

Вкладка	Назначение
Закладки	В этой вкладке отображается список закладок; чтобы открыть одну из них в главном окне, необходимо щелкнуть по ней левой кнопкой мыши
Журнал	Отображается журнал работы в Internet. Чтобы открыть один из показанных адресов в главном окне, достаточно щелкнуть по нему левой кнопкой мыши, чтобы удалить — щелкнуть по нему правой кнопкой мыши и в появившемся меню выбрать Удалить элемент, чтобы очистить журнал — выбрать пункт Очистить журнал. В этом же меню можно указать, в каком порядке следует отображать адреса в журнале — алфавитным или по датам. Если в том же меню выбрать пункт Настройка . . . , появится окно, в котором можно указать максимальный размер журнала, а также через какое время после последнего посещения адрес следует автоматически удалять. Если включить режим Расширенные подсказки, то будут отображаться более подробные сведения
Домашний каталог	Отображается домашний каталог; скрытые файлы (начинающиеся с точки) не отображаются. Чтобы открыть какой-либо файл в главном окне Konqueror, следует щелкнуть по нему левой кнопкой мыши. Если щелкнуть правой кнопкой мыши по каталогу, то с помощью появившегося меню можно открыть каталог в новом окне или в новой вкладке основного окна

Окончание табл. 4.7

Вкладка	Назначение
Сеть	Отображается список сетевых подключений (однако в нем также можно отображать и каталоги). Чтобы открыть объект в главном окне, достаточно щелкнуть по нему левой кнопкой мыши. Щелкнув по нему правой кнопкой мыши, можно вызвать меню, в котором будут предложены дополнительные возможности. Папки, отображаемые во вкладке Сеть, хранятся в каталоге <code>~/.kde/share/apps/konqsidebar/ virtual_folders/ remote/</code> . Создав в нем новый каталог, можно добавить во вкладку новую папку. Файлы, описывающие сетевые подключения, имеют расширение <code>.desktop</code> . Чтобы создать новый файл, используется пункт меню Создать \ Ссылку на Internet-адрес...
Корневой каталог	Корневой каталог — это начальная точка файловой системы; в нем хранятся все остальные файлы и каталоги. Путь к корневому каталогу обозначается косой чертой <code>«/»</code> . Развернув его, можно увидеть, в частности, каталог <code>root</code> , где хранятся файлы системного администратора, и <code>home</code> , в котором вы сможете найти свой домашний каталог
Инструменты	Эта вкладка предоставляет доступ к следующим инструментам: Обозреватель аудиокомпакт-диска; Устройства — отображается список имеющихся жестких дисков (и их разделов), дисководов и устройств чтения компакт-дисков. Чтобы подключить устройство и отобразить его содержимое в главном окне, надо щелкнуть по нему левой кнопкой мыши, чтобы отключить — щелкнуть правой кнопкой и в появившемся меню выбрать пункт Отключить; Обозреватель сети позволяет просматривать диски других компьютеров в локальной сети; Обозреватель системы печати предоставляет доступ к менеджеру печати KDE, Kprinter

Таблица 4.8. Настройка Панели навигации

Пункт меню	Возможности
Добавить	Добавить в панель навигации еще одну вкладку. Это может быть Панель проигрывателя (один из дополнительных модулей Konqueror) или какой-либо каталог
Несколько видов	Разделить панель навигации таким образом, что можно будет видеть две вкладки одновременно
Показывать вкладки слева	Этот пункт позволяет указать, с какой стороны панели отображаются значки вкладок
Показывать кнопку настройки	С помощью этого пункта можно указать, следует ли отображать Кнопку настройки
Щелкнув по значку какой-либо вкладки правой кнопкой мыши, можно вызвать меню, предлагающее следующие возможности	
Указать URL	Изменить путь (URL) к каталогу, который отображается в этой вкладке
Установить пиктограмму	Этот пункт позволяет сменить значок вкладки
Удалить	Удалить вкладку с панели навигации

Конqueror как менеджер файлов

В операционных системах Unix (и Linux) структура каталогов имеет форму нисходящего дерева: от главного (корневого) каталога ответвляются каталоги первого уровня, каждый из которых, в свою очередь, может содержать подкаталоги. Таким образом, из одного каталога в другой можно попасть, поднявшись «вверх» до общей точки, а далее — «вниз», углубляясь в подкаталоги, пока вы не дойдете до нужного (см. рис. 3.7).

Расположение любого файла или каталога в этой древовидной структуре можно описать с помощью «пути»: это список каталогов, которые необходимо пройти, чтобы найти нужный файл. Например, `/home/tata` означает подкаталог `tata` каталога `/home`, находящегося в корневом каталоге; а `/home/tata/words.txt` — файл `words.txt` в данном каталоге. Начальная косая черта («/») в приведенных путях означает корневой каталог.

Все имеющиеся файлы, включая те, которые находятся на других разделах жесткого диска, на гибком диске или на компакт-диске, доступны через корневой каталог, а их точное местоположение зависит от того, как настроена ваша система.

В системе Unix (или Linux) у каждого пользователя есть свой «домашний» каталог, в котором хранятся его личные файлы и настройки. Примеры таких каталогов — `/home/igor` или `/home/partyka`. Домашний каталог часто условно обозначают значком `~` (например, путь `~/letter.txt` указывает на файл `letter.txt` в домашнем каталоге).

Заметьте, что часто вместо термина «каталог» используют слово «папка», а также может встретиться термин «директорий» («директория»).

Кроме управления файлами, Конqueror позволяет также отображать в основном окне их содержимое (см. рис. 4.21 — отображение графического файла).

Режимы отображения. Когда Конqueror используется в качестве менеджера файлов, он показывает, какие файлы и подкаталоги находятся в определенном каталоге, а также предоставляет о них различные сведения.

Путь к текущему каталогу показан в заголовке окна и в строке адреса; заметьте, что в строке адреса путь приведен с префиксом `file:`, чтобы указать, что просматриваемый каталог — это часть обычной файловой системы локального компьютера (например, `file:/home/tata`), а не Internet-адрес.



Рис. 4.21. Отображение графического файла

Режим отображения файлов и папок может быть выбран с помощью меню Вид\Режим (табл. 4.9).

Таблица 4.9. Доступные варианты режимов отображения объектов

Режим	Пояснения
Значки	Для каждого файла отображается имя и соответствующий значок (см. рис. 4.17— 4.20)
Список	То же, что и режим Значки, однако все значки (и их имена) выстроены в столбцы
Таблица	Kонквеог показывает имя каждого файла (и небольшой значок) в отдельной строке. Также отображаются дополнительные сведения о файле (объем сведений можно изменить с помощью пункта меню Вид\Подробности)
Только текст	Похож на режим Таблица, но отличается тем, что не отображаются значки, а все имена каталогов начинаются с косой черты («/»)
Дерево (в виде дерева)	Этот режим удобен, когда необходимо просмотреть структуру каталогов, начиная с текущего. Файлы отображаются так же, как и в режиме Таблица, но слева от каждого имени каталога находится небольшой плюс + в квадратике. Щелкнув по нему, можно развернуть соответствующий каталог и увидеть его содержимое (рис. 4.22)
Таблица с подробностями	Этот режим похож на режим Таблица, однако отличается тем, что Конквеог по возможности показывает количество строк, слов и символов в каждом файле, а также формат файла

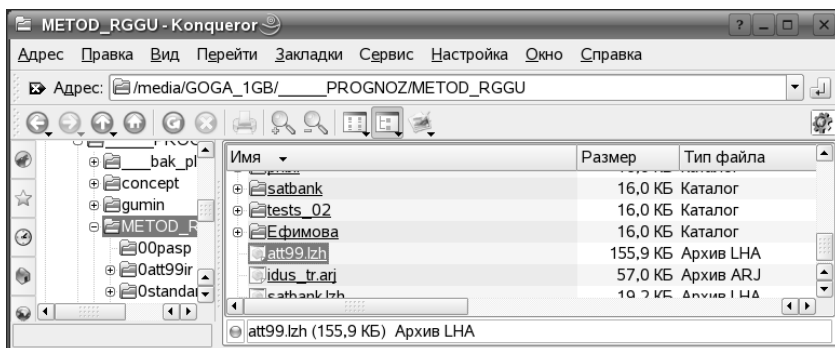


Рис. 4.22. Режим просмотра в виде дерева

Вместо стандартных значков (в режимах Значки и Список) Конквегор может показывать содержимое файла в уменьшенном виде.

Подсказки к файлам. Если включен режим Показывать подсказки к файлам в разделе Поведение окна настройки Конквегор, то при совмещении курсора мыши с именем файла (или значка), Конквегор будет показывать дополнительные сведения о файле в виде подсказки (рис. 4.23). Объем отображаемых сведений можно уточнить с помощью поля Количество строк в подсказках в окне настройки Конквегор.

Если включен режим Показывать миниатюры в подсказках, в окошке с подсказкой также будет отображено содержимое файла в уменьшенном виде.

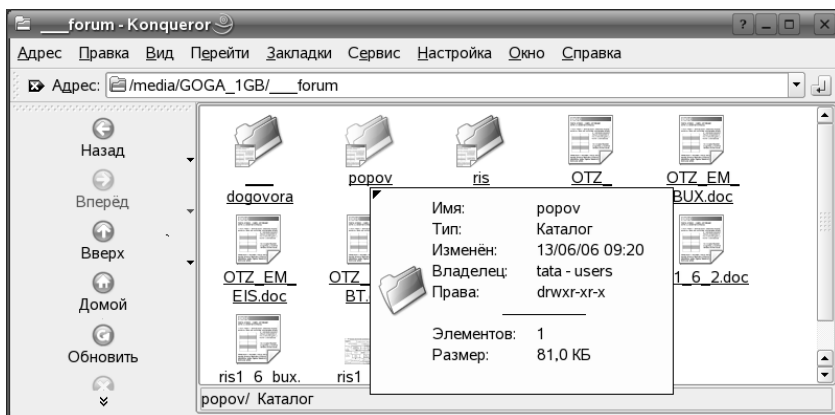


Рис. 4.23. Дополнительные сведения о файле

Просмотр миниатюр. Если выбрать пункт меню Вид\Миниатюры, активируется подменю, которое разрешает показ миниатюр для файлов определенного типа (вместо значка будет показано содержимое файла в уменьшенном виде). Если просмотр миниатюр включен для звуковых файлов, то файлы будут воспроизведены.

Просмотр миниатюр доступен только в режимах Значки и Список.

Для того чтобы создать миниатюру файла, Konqueror необходимо прочесть гораздо больше данных, чем только имя файла; поэтому при просмотре файлов на гибком диске или в сети бывает удобно отключить миниатюры. Воспользовавшись разделом Миниатюры в окне настройки Konqueror, можно отключить показ миниатюр для таких протоколов, как FTP, так как иначе чтение дополнительных данных будет отнимать слишком много времени.

Перемещение по каталогам. Чтобы найти нужный файл, необходимо открыть каталог, в котором он находится. Переходя от одного каталога к другому, можно перемещаться по древовидной структуре каталогов, а именно:

- для перехода в подкаталог следует щелкнуть по нему левой кнопкой мыши, а если он уже выделен (см. ниже), нажать клавишу <Enter>;
- чтобы перейти в родительский каталог, необходимо нажать кнопку Вверх на панели инструментов. Также можно использовать комбинацию клавиш <Alt+↑> или же пункт меню Переход\Вверх;
- чтобы выделить файл или папку, не открывая ее, следует использовать клавиши со стрелками. Сведения о выделенном объекте будут показаны в строке состояния;
- чтобы облегчить обзор каталогов, можно воспользоваться режимом Древа — в этом режиме слева от каждого каталога появляется небольшой квадратик. Если в нем показан знак , щелчок по нему позволит развернуть данный каталог, т. е. отобразить все его файлы и подкаталоги в виде ветвей. Знак в квадратике сменится на , щелчок по нему позволяет свернуть каталог.

Боковая панель навигации также может использоваться для поиска необходимого каталога.

К известному каталогу можно перейти следующими способами:

- введя путь в строке адреса;

- выбрав пункт меню Адрес\Открыть страницу;
- нажав <Ctrl+O>.

Следует помнить, что в Unix (и в Linux) в названиях файлов строчные и прописные буквы различаются.

Для перехода в подкаталог необходимо выполнить одно из следующих действий:

- нажать кнопку Вперед на панели инструментов;
- выбрать пункт меню Переход\Вперед;
- или нажать <Alt+→>.

Наоборот, можно выйти из подкаталога следующим образом:

- нажав кнопку Назад на панели инструментов;
- выбрав пункт меню Переход\Назад;
- нажав <Alt+←>.

Подсказка. При нажатии и удержании нажатой левой кнопки мыши на значке Вверх, Назад или Вперед появится меню, из которого могут быть выбраны последние посещенные адреса.

Поиск файлов и каталогов. Для поиска необходимого файла, следует нажать кнопку Поиск файла на панели инструментов или выбрать пункт меню Сервис\Поиск файла.... В окне Konqueror появится окошко программы KFind.

Если название файла или каталога начинается с точки, он считается скрытым, и Konqueror его обычно не показывает. Чтобы изменить этот режим, следует установить флажок напротив пункта меню Вид\Показывать скрытые файлы. Konqueror также может не показывать некоторые файлы, если включен фильтр файлов; при этом будут отображаться только файлы определенного типа.

Сменные накопители. Имеющиеся на компьютере разделы жесткого диска, гибкие диски, компакт-диски и флэш-накопители обычно доступны через один из подкаталогов в каталоге /, /mnt или /media (рис. 4.24). Например /media/floppy или media/cdrom (точное местоположение каталогов зависит от того, как настроена операционная система).

В Unix (и Linux) перед тем, как работать с гибким диском (компакт-диском или флэш-памятью), его необходимо подключить командой `mount` с консоли (например, `mount /mnt/floppy`), если необходимо получить доступ к другим разделам жесткого диска, их также следует подключить (рис. 4.25).

Перед тем как вынимать диск из дисковод, наоборот, его необходимо отключить (командой `umount`, например, `umount /mnt/floppy`, или командой Отключить из контекстного меню

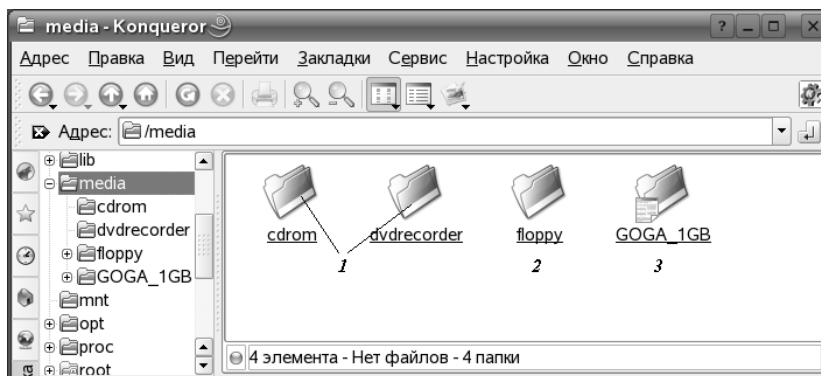


Рис. 4.24. Содержание каталога media:

1 — оптические накопители; 2 — гибкий диск; 3 — флэш-память

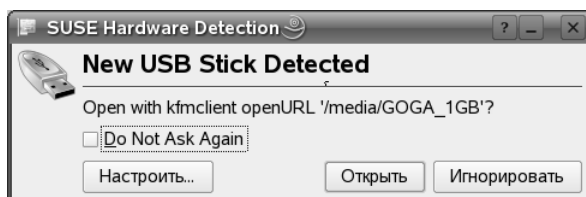


Рис. 4.25. Окно диалога подключения накопителя

раздела на рабочем столе). Если же в системе поддерживается автоматическое подключение (automount), отсутствует необходимость вручную подключать и отключать устройства.

Если на рабочем столе есть значки разделов, достаточно щелкнуть по одному из них, тогда диск будет автоматически подключен и открыт в окне Konqueror. Чтобы отключить диск, следует щелкнуть по значку правой кнопкой мыши, и в появившемся меню выбрать пункт Отключить. Создание таких значков осуществляется в разделе меню Создать.

Вместо того чтобы открывать терминал-консоль, можно также воспользоваться в Konqueror пунктом меню Сервис\Выполнить команду оболочки (<Ctrl+E>).

Удаление файлов и каталогов. Для того чтобы удалить файл или каталог, пользователь должен иметь соответствующие права. Konqueror предоставляет два способа удаления файла или каталога:

- перенести его в корзину (нажатием клавиши <Del>), что является логическим удалением, так как если файл был удален по ошибке, его можно восстановить;

- физически удалить его (нажатием <Shift+Del>), при этом файл будет удален из каталога, а пространство, которое он занимал на диске, станет доступно для размещения других файлов. Это также можно сделать с помощью команды `rm`.

Другой способ удалить файл или каталог — щелкнуть по нему правой кнопкой мыши и в появившемся меню выбрать один из следующих пунктов: Выбросить в корзину, Удалить или Уничтожить. Если нужный файл или каталог уже выделен, также можно воспользоваться одним из пунктов меню Правка: Выбросить в корзину, Удалить или Уничтожить.

Копирование и перемещение. Для того чтобы скопировать или переместить файл (или каталог), пользователь должен иметь соответствующие права. Копировать или переместить файл (каталог) можно несколькими способами:

- в контекстном меню выбрать пункт Копировать. Если нужный файл или каталог уже выделен, можно также воспользоваться кнопкой Копировать на панели инструментов, пунктом меню Правка\Копировать или сочетанием клавиш <Ctrl+C>. После этого необходимо перейти в каталог, в который должен быть помещен файл, и нажать кнопку Вставить на панели инструментов. Вставку можно осуществить, также выбрав пункт меню Правка\Вставить или нажав <Ctrl+V> либо же поместив указатель мыши в свободную область окна, щелкнуть правой кнопкой мыши и в появившемся меню выбрать пункт Вставить;
- перемещение файла осуществляется так же, как и копирование, за исключением того, что вместо пункта меню Правка\Копировать (кнопки Копировать) следует воспользоваться пунктом (или кнопкой) Вырезать или комбинацией клавиш <Ctrl+X>. При вставке вырезанный файл будет перемещен в данный каталог;
- копировать или переместить выделенные файлы можно также, выбрав пункт меню Правка\Копировать файлы (<F7>), Правка\Переместить файлы (<F8>) или щелкнув по файлу правой кнопкой мыши и в появившемся меню выбрав пункт Копировать в ... либо Переместить в

«Перетаскивание» мышью. Konqueror также поддерживает копирование и перемещение файлов с помощью мыши.

Для этого открывают два окна (или вкладки) Konqueror: одно — с каталогом, где находится исходный файл, а другое — с

каталогом, куда этот файл необходимо скопировать. Щелкнув по файлу левой кнопкой мыши и удерживая ее нажатой, следует передвинуть курсор мыши во второе окно, на место, не занятое другими значками, и отпустить кнопку мыши. После этого в появившемся меню необходимо выбрать пункт Копировать или Переместить. Если значок «перенесен» не на свободное место, а на значок каталога, файл будет помещен в этот, а не в текущий каталог.

Можно настроить Konqueror таким образом, что в одном окне будет отображаться сразу несколько каталогов, что сделает перетаскивание мышью гораздо удобнее.

На рис. 4.26 показано, что окно можно разделить с помощью пункта меню *Окно\Разделить панель* по вертикали или комбинации клавиш *<Ctrl+Shift+L>*, при этом в каждом окне будет показан свой каталог.

Чтобы Konqueror мог отображать в обеих панелях разные каталоги, необходимо убедиться, что панели не связаны между собой — квадратик в нижнем правом углу каждой из них должен быть пуст.

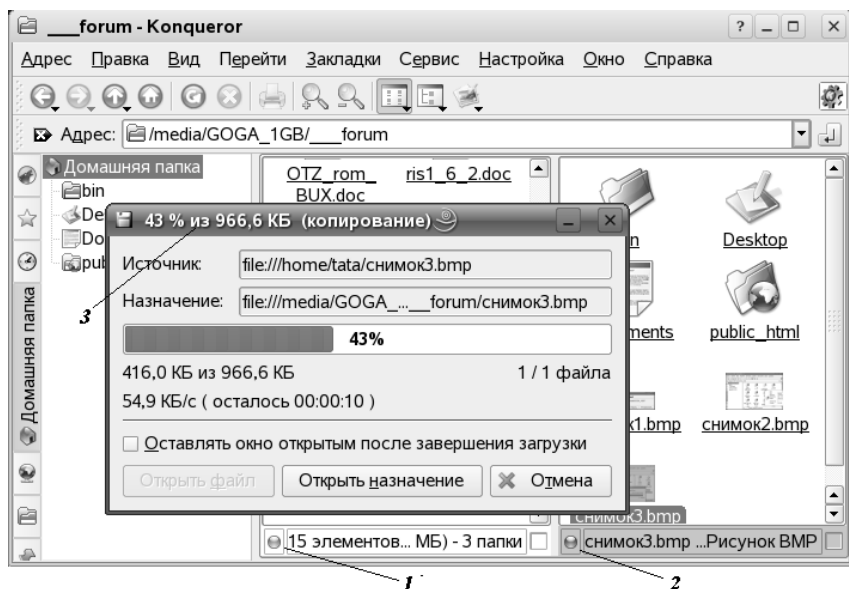


Рис. 4.26. Разделение окна для копирования файлов:

1 — активное окно; 2 — пассивное окно; 3 — окно протокола копирования файла

В каждый момент активна только одна из панелей — слева на строке состояния здесь будет расположен зеленый индикатор. Чтобы перейти к другой панели (сделать ее активной), следует щелкнуть по любому свободному месту в самой панели или в ее строке состояния.

Чтобы убрать текущую панель, можно воспользоваться комбинацией клавиш <Ctrl+Shift+R> или пунктом меню *Окно\Удалить активную панель* либо же щелкнуть правой кнопкой мыши по строке состояния этой панели и в появившемся меню выбрать пункт *Удалить активную панель*.

Работая со вкладками в Konqueror, можно перетащить файл на заголовок вкладки, не отпуская кнопку, тогда она откроется и можно будет перенести файл куда следует.

Файлы с одинаковыми именами. При попытке вставить файл в каталог, в котором файл с таким именем уже имеется, Konqueror предложит следующие действия (рис. 4.27):

- **Заменить** — новый файл будет записан поверх старого. Если копируются несколько файлов, также можно выбрать *Заменить все*;
- **Пропустить** — файл скопирован не будет. При копировании нескольких файлов также можно выбрать *Пропустить все*;
- **Дать копируемому файлу новое имя** — можно ввести имя вручную или позволить это сделать Konqueror, нажав на кнопку *Предложить*, затем на кнопку *Переименовать*.

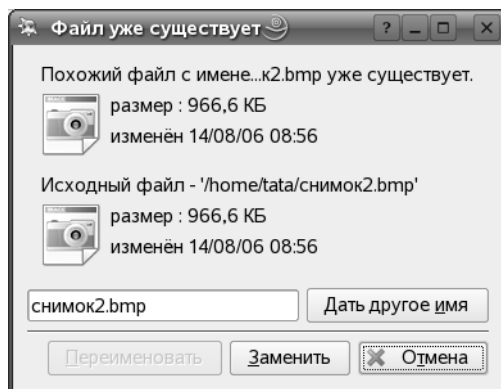


Рис. 4.27. Попытка копирования файла на место существующего

Выделение нескольких файлов. Иногда бывает необходимо скопировать, переместить или удалить несколько файлов, которые имеют между собой что-то общее, например перенести все графические файлы с расширением `.png` из одного каталога в другой. Konqueror облегчает эту задачу, позволяя автоматически выделить группу файлов, имеющих похожие имена.

С этой целью следует выбрать пункт меню `Правка\Выделить...` или нажать `<Ctrl++>`, и в появившемся окне ввести необходимый шаблон файла (используя знак `*`, означающий любое количество символов, или `?`, означающий один символ). Если нажать кнопку `OK`, то Konqueror автоматически выделит все файлы, имена которых совпадают с данным шаблоном.

Например:

- при шаблоне `flag*.png` будут выделены все имена файлов, начинающиеся с подстроки `flag` и заканчивающиеся `.png`;
- при шаблоне `memo?.txt` будут выделены, к примеру, файлы `memo1.txt` и `memo9.txt`, но не `memo99.txt`.

Можно уменьшить количество выделенных файлов, указав, с каких файлов следует снять выделение, для чего следует выбрать пункт меню `Правка\Отменить выделение...` или нажать `<Ctrl+\`.

Чтобы снять выделение со всех файлов, необходимо нажать `<Ctrl+U>`, выбрать пункт меню `Правка\Отменить все выделение` или щелкнуть по любому месту окна, не занятому значками. Имеется также возможность обратить выделение, т. е. снять его со всех выделенных файлов и выделить остальные, для чего необходимо выбрать пункт меню `Правка\Инвертировать выделение` или нажать `<Ctrl+*>`.

В табл. 4.10 приведены примеры клавиш, которые часто используются для выделения файлов в режимах `Таблица`, `В виде дерева` и `Только текст`.

Когда необходимые файлы выбраны, могут использоваться такие команды, как копирование, перемещение или удаление — их действие будет распространяться на все файлы сразу.

В зависимости от установленной клавиатуры и настройки системы для сочетаний `<Ctrl++>`, `<Ctrl+\` и `<Ctrl+*>`, возможно, потребуется использовать клавиши `<+>`, `<->` и `<*>` на цифровой части клавиатуры (см. рис. 1.25, 6).

Таблица 4.10. Клавиши, используемые для выделения файлов

Клавиша	Действие
<Space>, пробел	Выделить или снять выделение
<Ins>	Обратить выделение и перейти к следующему файлу
<Ctrl+↑>, <Ctrl+↓>, <Ctrl+Home>, <Ctrl+End>, <Ctrl+PgUp>, <Ctrl+PgDn>	Перейти в соответствующем направлении, по пути обращая выделение всех файлов
<Shift+↑>, <Shift+↓>, <Shift+Home>, <Shift+End>, <Shift+PgUp>, <Shift+PgDn>	Снять все выделение, затем перейти в соответствующем направлении, по пути выделяя все файлы

Меню «Создать». В режиме менеджера файлов можно выбрать пункт Создать новый... из меню Правка или щелкнуть правой кнопкой мыши по любому месту окна, которое не занято файлами. Появится меню, с помощью которого можно создать в текущем каталоге ряд объектов (табл. 4.11).

Таблица 4.11. Основные пункты меню Создать

Пункты меню	Действия
Файл\Ссылка на приложение...	Можно создать значок, щелкнув по которому будет запускаться определенное приложение. При выборе этого пункта появится окно с тремя вкладками: Общие — здесь можно выбрать значок приложения и указать его название; Права доступа — указать, кто имеет право использовать или изменять создаваемую ссылку; Запуск — указать путь к запускаемой программе, например kedit, чтобы запустить редактор KEdit, или kedit /home/tata/todo.txt, чтобы запустить KEdit и открыть в нем файл /home/tata/todo.txt. Если необходимо поместить значок на рабочий стол, надо создать ссылку в каталоге ~/Desktop (в зависимости от настройки системы этот файл может называться и по-другому). Также можно щелкнуть по рабочему столу правой кнопкой мыши и воспользоваться таким же меню Создать. Если требуется создать множество ссылок, и вы не хотите, чтобы они загромождали рабочий стол, можно поместить их в специальный каталог для приложений. Чтобы перейти в этот каталог, следует выбрать пункт меню Переход\Приложения. После этого останется лишь поместить ссылку на этот каталог (обычно это ~/.kde/share/applink) на рабочий стол. Если необходимо, чтобы ссылка на приложение находилась на панели быстрого запуска, следует создать ссылку в каталоге приложений и перетащить ее мышью на любое свободное место на панели

Окончание табл. 4.11

Пункты меню	Действия
Файл\Ссылка на URL...	Можно создать значок, щелкнув по которому осуществляется переход к определенному адресу Internet или к каталогу. Как и в случае с пунктом Ссылка на приложение..., можно поместить значок на рабочий стол, если создать ссылку в каталоге ~/Desktop или использовать меню Создать... прямо на рабочем столе (щелкнув по нему правой кнопкой мыши). После создания значка в качестве его имени будет отображаться адрес, на который он ссылается. Чтобы изменить имя, надо щелкнуть по значку правой кнопкой мыши и в появившемся меню выбрать пункт Свойства..., в окне свойств выбрать вкладку Общие и ввести новое название значка
Устройство\Ссылка на гибкий диск...	Пункт позволяет создать значок, щелкнув по которому можно просмотреть содержимое гибкого диска; диск будет также автоматически подключен. Чтобы отключить диск после завершения работы с ним, в контекстном меню его значка следует выбрать пункт Отключить. Это также применимо для значка жесткого диска, любых других разделов. Чаще всего такие значки размещают на рабочем столе
Ссылка на жесткий диск...	Пункт позволяет выполнить те же действия, что и Ссылка на гибкий диск..., но с жестким диском или его разделом
Устройство CD/DVD-ROM...	Этот пункт позволяет выполнять те же действия, что и Ссылка на гибкий диск..., но с компакт-диск или DVD
Каталог...	Этот пункт позволяет создать (под)каталог
Текстовый файл...	Этот пункт позволяет создать обычный текстовый файл с заданным именем
Файл HTML...	Этот пункт позволяет создать файл HTML, при именовании которого желательно в качестве расширения указать .html, чтобы избежать возможных недоразумений
Файл презентации...	Этот пункт позволяет создать файл презентации, с которым можно будет работать в KPresenter (желательно в качестве расширения указать .kpr)
Текстовый документ...	Этот пункт позволяет создать документ, с которым можно будет работать в KWord. Когда вы будете давать ему имя, желательно в качестве расширения указать .kwd
Документ электронной таблицы...	Этот пункт позволяет создать электронную таблицу, с которой можно будет работать в KSpread (желательно в качестве расширения указать .ksp)
Документ иллюстратора...	Этот пункт позволяет создать файл, с которым можно будет работать в Kontour (желательно в качестве расширения указать .kil)

Переименование и изменение прав доступа к файлу. Самый простой способ переименовать файл или каталог — это щелкнуть по его значку правой кнопкой мыши и в появившемся меню выбрать пункт Переименовать.

Чтобы изменить права доступа к файлу или каталогу, следует щелкнуть по его значку правой кнопкой мыши и в появившемся меню выбрать пункт *Свойства...* или, если файл уже выделен, воспользоваться пунктом меню *Правка\Свойства*. Появится окно свойств файла, содержащее несколько вкладок:

- вкладка *Общие* предоставляет общие сведения об объекте и позволяет изменить его имя и, в случае с каталогом, его значок;
- Вкладка *Права* позволяет изменить права доступа к объекту.

Работа в режиме администратора. Если вы зашли в систему как обычный пользователь и пытаетесь получить доступ к файлам вне вашего домашнего каталога, вы нередко будете получать сообщение об ошибке: *Отказано в доступе*.

Чтобы получить доступ ко всем файлам, необходимо зайти в систему под именем администратора, или *root*. Вместо того чтобы выходить из системы, а затем вновь заходить, можно запустить Konqueror сразу в режиме администратора. Для этого в главном меню *К* выберите пункт *Система\Менеджер файлов* (в режиме администратора).

Работа с командной строкой. Для вызова режима командной строки можно, например, запустить *Konsole*, воспользовавшись пунктом меню *Сервис\Открыть терминал* или нажав *<Ctrl+T>*. Если необходимо лишь запустить программу или открыть документ, достаточно воспользоваться пунктом меню *Сервис\Выполнить команду...* (*<Alt+F2>*).

Если выбрать пункт меню *Сервис\Выполнить команду оболочки...* (*<Ctrl+E>*), появится окно, в котором можно ввести команду. Следует отметить, что в этом случае вам будут недоступны некоторые возможности (например, не будут корректно работать программы *top* и *less*), однако преимущество данного способа в том, что не придется тратить время на запуск *Konsole*.

Для более сложных действий в Konqueror имеется еще одна возможность — если выбрать пункт меню *Окно\Показать эмулятор терминала*, то будет открыто окно терминала прямо в Konqueror в виде еще одной панели. Кроме того, если в правом нижнем углу одной из обычных панелей установлен специальный флажок, то какой бы ни был выбран каталог, в терминале он также станет текущим (рис. 4.28).

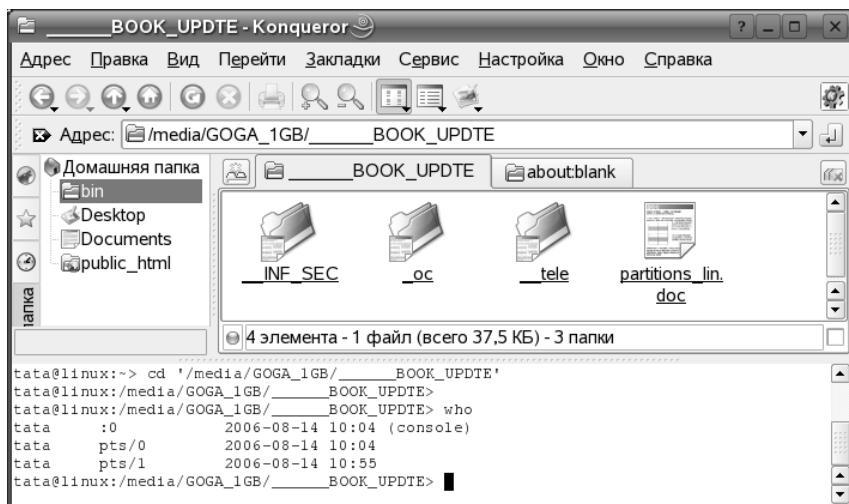


Рис. 4.28. Konqueror с открытым экраном эмулятора терминала

Подключение к Internet. Как только будет установлено соединение с Internet, появляется возможность просмотра Web-страниц так же, как и файлов на локальном компьютере.

Если подключение к Internet осуществляется с помощью модема, установка связи реализуется программой KPPP или другой программой дозвона.

Если компьютер подключен к Internet через локальную сеть и необходимо использовать прокси-сервер, можно указать его адрес в разделе Прокси-сервер окна настройки Konqueror (окно можно вызвать, выбрав пункт меню Настройка\Настроить Konqueror...).

Если же компьютер подключен к Internet по высокоскоростному каналу, в таком случае настройка программного обеспечения зависит от оператора, а также от настройки операционной системы.

Konqueror в режиме Web-обозревателя

Работа с Internet и поиск. При нахождении в Internet можно просматривать Web-страницы в Konqueror так же, как в любом другом обозревателе.

На рис. 4.29 приведен пример отображения содержания нескольких последовательных страниц в окнах Konqueror.

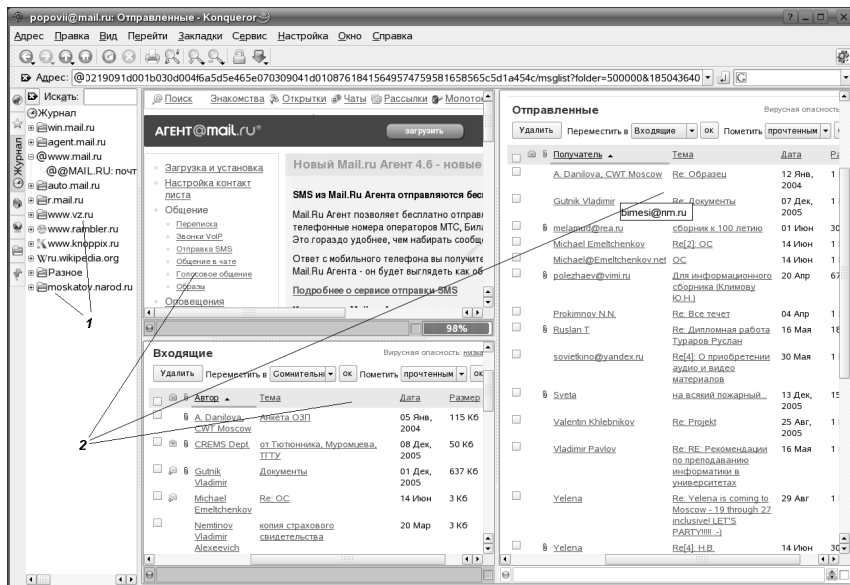


Рис. 4.29. Отображение HTML-файлов:
1 — панель навигации; 2 — Web-страницы

Необходимо ввести адрес страницы в строку адреса, нажать клавишу <Enter> или кнопку  (Переход) справа от строки адреса, и Konqueror загрузит и отобразит указанную страницу. Сообщение Узел неизвестен обычно означает, что не установлена связь с Internet (или указан неверный адрес страницы).

Если эта страница ранее посещалась, Konqueror облегчит ввод адреса с помощью функции автозавершения текста. Имеется также возможность выбрать нужный адрес из списка посещенных страниц, для этого следует воспользоваться разделом Журнал на панели навигации. Если необходимо произвести поиск в Internet, специальные Web-сокращения помогут легко указать нужную поисковую машину. В табл. 4.12 приводятся основные действия при навигации по Web-страницам.

Работа с вкладками. При работе с одним окном Konqueror можно переключаться между Web-страницами/каталогами посредством вкладок. Например, можно открыть страницу «на фо-

Таблица 4.12. Команды навигации Konqueror

Операция	Действия
Переход по ссылке	Щелкнуть по ссылке левой кнопкой мыши
Открыть ссылку в новом окне, не закрывая старого	Щелкнуть по ссылке средней кнопкой мыши. Также можно щелкнуть по ней правой кнопкой мыши и в появившемся меню выбрать пункт Открыть в новом окне
Разделить окно на несколько панелей, в каждой из которых будет отображаться отдельная страница	Сочетание клавиш <Ctrl+Shift+I> либо пункт меню Окно\Разделить панель по вертикали. Konqueror позволяет также работать со всеми страницами в одном окне, не разделяя окно на панели
Переход к предыдущей странице	Сочетание клавиш <Alt+←>, или кнопка  (Назад) на панели инструментов, или пункт меню Переход\Назад
Переход к следующей странице	Сочетание клавиш <Alt+→>, кнопка  (Вперед) на панели инструментов или пункт меню Переход\Вперед
Остановить загрузку страницы	Нажать клавишу <Esc>, кнопку  (Остановить) на панели инструментов или выбрать пункт меню Вид\Остановить
Увеличить или уменьшить размер изображения текста и рисунков	Воспользоваться кнопками  (+) или  (-) на панели инструментов

новой вкладке», чтобы перейти к ней позже, когда она полностью загрузится.

Чтобы включить этот режим, следует щелкнуть по ссылке, которую необходимо открыть правой кнопкой мыши и в появившемся меню выбрать пункт Открыть в новой вкладке. Страница будет открыта в том же окне Konqueror, но для нее будет создана новая вкладка (появляется список вкладок сразу над страницей). Чтобы перейти к какой-либо странице, необходимо щелкнуть по ее вкладке или использовать сочетания клавиш <Ctrl+[> и <Ctrl+]>, которые позволяют перейти к предыдущей или последующей странице в окне.

Если щелкнуть по ссылке правой кнопкой мыши и в появившемся меню выбрать пункт Открыть в фоновой вкладке, страница будет открыта в отдельной вкладке, но в фоновом режиме: пока она загружается, вы сможете продолжать работу с текущей страницей. Чтобы перейти к открытой вами странице, щелкните мышью по ее вкладке.

Если в окне настройки Konqueror (Настройка\Настроить Konqueror..., раздел Поведение) включить режим Открывать

ссылки в новых вкладках, а не в новых окнах, то при щелчке на любой ссылке средней кнопкой мыши страница будет открываться в новой вкладке того же окна. Если же при щелчке средней кнопкой мыши удерживать нажатой клавишу <Shift>, страница будет открыта в фоновом режиме.

Если щелкнуть по вкладке правой кнопкой мыши, то появится меню операций над вкладками (табл. 4.13).

Таблица 4.13. Меню операций над вкладками

Пункт меню	Назначение
Новая вкладка	Открыть в окне Konqueror новую вкладку. Чтобы просмотреть в этой вкладке какую-либо страницу, надо ввести Internet-адрес в строку адреса, выбрать одну из закладок на панели закладок или воспользоваться журналом на панели навигации
Создать копию вкладки	Создать новую вкладку, идентичную выбранной
Отделить вкладку	Открыть выбранную вкладку в новом окне Konqueror, одновременно удаляя ее из текущего
Заккрыть вкладку	Этот пункт позволяет закрыть выбранную вкладку
Обновить	Загрузить содержание вкладки заново
Обновить все вкладки	Обновить содержимое всех закладок
Перейти к вкладке	Подменю со списком всех закладок. Выбор вкладки в этом списке делает ее активной
Заккрыть другие вкладки	Этот пункт позволяет закрыть все вкладки, кроме выбранной

Web-сокращения. В Konqueror имеется возможность использовать Web-сокращения — сокращенные названия поисковых машин, с помощью которых поиск в Internet активизируется вводом искомой фразы в строке адреса (без захода на главную страницу поисковой машины).

Например, если ввести в строке адреса `gg:konqueror` и нажать <Enter>, обозреватель автоматически передаст поисковой машине Google запрос показать информацию об обозревателе Konqueror, которая есть в Internet.

Чтобы просмотреть список доступных Web-сокращений или добавить новые, следует выбрать пункт меню *Настройка\Настройка Konqueror...*, в появившемся окне войти в раздел Настройки и щелкнуть по значку Web-сокращения.

Идентификация обозревателя. Когда Konqueror подключается к какому-либо серверу в Internet, он сообщает туда некоторые собственные данные (название, версия, система и т. д. или «User Agent»), исходя из которых сервер пересылает обозревателю такой вариант страницы, который в Konqueror будет выглядеть наилучшим образом. Это связано с тем, что у каждого обозревателя есть свои особенности, и не во всех обозревателях одна и та же страница будет выглядеть одинаково.

Однако, встречаются серверы, настроенные таким образом, что тем обозревателям, которые их администраторы считают «неправильными», вообще запрещается доступ (несмотря на то что, к примеру, Konqueror большинство страниц отображает без каких бы то ни было ошибок).

Чтобы решить эту проблему, Konqueror позволяет изменить сведения о себе, которые он передает определенному серверу или всем серверам заданной зоны (домена). Для этого следует выбрать пункт меню *Настройка\Настроить Konqueror...*, в появившемся окне войти в раздел *Настройка* и щелкнуть по значку *Псевдонимы*.

Если страница все же отображается неверно, целесообразно проверить, включена ли поддержка приложений Java и JavaScript. Для этого в окне настройки Konqueror необходимо выбрать раздел Java и JavaScript и убедиться, что поддержка включена.

Сохранение и печать страниц. Если необходимо сохранить текущую Web-страницу на жестком диске, следует выбрать пункт меню *Адрес\Сохранить как...* При наличии на странице врезок (frames), каждая из них может быть сохранена отдельно, если щелкнуть по ней левой кнопкой мыши и выбрать пункт меню *Адрес\Сохранить врезку как...* В табл. 4.14 приводятся различные варианты команд сохранения и печати страниц.

Работа с протоколом FTP. FTP (file transfer protocol — протокол передачи файлов) является одним из самых ранних способов передачи файлов с одного компьютера на другой по Internet. FTP позволяет работать с файлами и каталогами удаленного компьютера так, как если бы они находились на локальном жестком диске.

Можно осуществлять копирование файлов с сервера на локальный компьютер с помощью команд *Копировать* и *Вставить*, «перетаскивать» файлы мышью и, если у пользователя есть соответствующие права, передавать файлы с компьютера на сервер.

Таблица 4.14. Сохранение и печать Web-страниц

Действие	Команда
Сохранить фоновый рисунок страницы	Выбрать пункт меню Адрес\Сохранить фоновый рисунок как...
Сохранить какой-либо из рисунков на самой странице	Щелкнуть по рисунку правой кнопкой мыши и в появившемся меню выбрать пункт Сохранить рисунок как...
Сохранить на локальном компьютере файл, на который указывает ссылка	Щелкнуть правой кнопкой мыши по ссылке (рисунок также может быть ссылкой) и в появившемся меню выбрать пункт Сохранить конечный документ как...
Скопировать ссылку в буфер обмена (после чего можно вставить ее, например, в письмо и послать по электронной почте)	Щелкнуть по ссылке правой кнопкой мыши и в появившемся меню выбрать пункт Копировать ссылку
Скопировать в буфер обмена адрес рисунка	Щелкнуть по рисунку правой кнопкой мыши и в появившемся меню выбрать пункт Копировать адрес рисунка
Сохранить страницу целиком, включая рисунки	Выбрать пункт меню Сервис\Сохранить в архив... Страница будет сохранена в файл с расширением .war (эта возможность предоставляется дополнительным модулем Konqueror и поэтому может быть недоступна)
Напечатать страницу на принтере	Выбрать пункт меню Адрес\Печать... или Печать врезки, или воспользоваться кнопкой  (Печать) на панели инструментов

Например, если ввести в строку адреса текст

`ftp://ftp.kde.org/pub/kde`

и нажать клавишу <Enter>, то увидим содержимое каталога /pub/kde на сервере `ftp://ftp.kde.org/pub/kde/`.

Строго говоря, хотя все адреса FTP-серверов должны начинаться с `ftp://`, а адреса Web-страниц — с `http://`, Konqueror обычно позволяет опускать эти префиксы.

Для доступа к FTP-серверу обычно необходимо ввести имя пользователя и пароль. Однако большинство серверов, которые позволяют бесплатно загружать с них файлы, принимают в качестве имени пользователя слово `anonymous` («анонимный»), а в качестве пароля — адрес электронной почты пользователя (см. также рис. 4.6). При подключении к серверу Konqueror автоматически передает серверу эти данные, избавляя от необходимости каждый раз вводить их самостоятельно.

Если сервер позволяет вход под именем `anonymous`, но сообщает, что в данный момент он перегружен, Konqueror может воспринять этот ответ как сообщение о неверном имени или пароле и предложит ввести их заново.

Если сервер предоставляет доступ только зарегистрированным пользователям, потребуется ввести имя и пароль — можно либо подождать, пока Konqueror их запросит, либо ввести имя вместе с адресом, например:

```
ftp://username@ftp.cia.org
```

В Konqueror также имеется возможность автоматически передавать серверу имя и пароль, которые указаны в файле `.netrc`.

Автоматическое завершение текста. В Konqueror имеется возможность, которая существенно облегчает ввод Internet-адресов — автозавершение текста. Чтобы включить автозавершение, необходимо щелкнуть правой кнопкой мыши по свободной части строки адреса и в появившемся меню выбрать пункт Автозавершение текста. Подменю содержит следующие варианты (табл. 4.15).

Таблица 4.15. Варианты режима Автозавершение текста

Режим	Действия
Выключено	Автозавершение выключено
Вручную	После ввода часть адреса можно нажать <Ctrl+E>, и Konqueror допишет недостающую часть адреса (если это возможно)
Автоматически	По мере ввода адреса Konqueror предлагает варианты окончания адреса (добавляемая часть будет выделена). Если вариант не устраивает, следует продолжать вводить адрес вручную, если предложен правильный вариант, нажать клавишу <Enter>
Выбор из списка	По мере ввода адреса Konqueror отображает выпадающий список, в котором показаны варианты полного адреса. Если один из вариантов подходит, следует щелкнуть по нему левой кнопкой мыши, также можно выделить его с помощью клавиш <↓> или <↑>, а затем нажать <Enter>
До косой черты	Похоже на режим Автоматически, однако адрес дополняется только до следующей косой черты (/). Если предлагаемый вариант устраивает, следует нажать <Ctrl+E>, когда будет введен весь адрес, нажать <Enter>

Журнал ранее посещенных адресов, который Konqueror использует для автозавершения, можно просмотреть и изменить на странице Журнал панели навигации.

Работа с закладками. Закладки позволяют сохранить адреса часто посещаемых страниц в Internet, а также файлов и каталогов. Закладки также доступны с панели навигации.

Чтобы открыть список закладок, необходимо выбрать пункт меню Закладки или нажать <Alt+З> и после появления списка, выделив необходимый пункт с помощью клавиш со стрелками, нажать <Enter>.

Создать новую закладку можно следующими способами:

- выбрать пункт меню Закладки\Добавить закладку;
- нажать сочетание клавиш <Ctrl+B>;
- щелкнуть по свободной части страницы правой кнопкой мыши и в появившемся меню выбрать пункт Добавить в закладки.

Список закладок может содержать папки, в которых находятся другие закладки. Чтобы создать папку, следует выбрать пункт меню Закладки\Создать папку.... После этого, если необходимо добавить какую-либо закладку не в основной список, а в эту папку, надо выбрать папку в меню Закладки и воспользоваться пунктом Добавить закладку.

Редактор закладок. При выборе пункта меню Закладки\Редактор закладок, появляется окно Редактора закладок.

В окне представлены закладки и папки в виде дерева. Слева от каждой папки будет находиться квадратик со знаком «+», при щелчке по которому папка развернется, и будут показаны содержащиеся в ней подкаталоги. При этом знак в квадратике сменится на «-» и, если щелкнуть по нему еще раз, папка будет свернута.

Чтобы выбрать нужную папку, достаточно щелкнуть по ней мышью или воспользоваться клавишами <↑>, <↓>. Клавиша <→> позволяет развернуть папку, а <←> — свернуть.

Закладки и папки можно перетаскивать мышью, а также перемещать с помощью команд Вырезать и Вставить. В главном меню закладки будут располагаться точно в таком же порядке, как и в данном окне. Чтобы вставить в меню разделительную черту, следует воспользоваться пунктом меню Вставка\Вставить разделитель.

Чтобы создать новую папку, необходимо выбрать пункт меню Вставка\Создать новую папку, щелкнуть правой кнопкой мыши по любому объекту главного окна редактора и в появившемся окне выбрать пункт Создать новую папку или воспользоваться сочетанием клавиш <Ctrl+N>.

Чтобы дать закладке или папке новое имя, ее необходимо выделить, а затем нажать клавишу <F2> или выбрать пункт меню Правка\Переименовать. Можно также щелкнуть по закладке правой кнопкой мыши и в появившемся меню выбрать пункт Переименовать. Чтобы изменить адрес страницы, на которую указывает закладка, следует нажать <F3> или выбрать пункт меню Изменить адрес.

Редактор закладок позволяет импортировать закладки из других обозревателей, добавляя их в новую папку или замещая ими уже существующие. Чтобы воспользоваться этой возможностью, надо выбрать пункт меню Файл\Импорт. С помощью пункта меню Файл\Экспорт закладки можно экспортировать в обозреватели Netscape или Mozilla.

Если наряду с Konqueror часто используется обозреватель Netscape, то может оказаться удобным включить режим Показывать закладки Netscape в окнах Konqueror в меню Настройка. В этом случае при добавлении или удалении закладки в Netscape она также будет добавлена или удалена в Konqueror.

Чтобы указать, какая именно папка отображается как панель закладок, необходимо ее выделить, а затем выбрать пункт меню Правка\Сделать папку панелью закладок.

Если требуется узнать, на что ссылается определенная закладка, достаточно щелкнуть по ней правой кнопкой мыши и в появившемся меню выбрать пункт Открыть в Konqueror. Если необходимо лишь убедиться, что адрес все еще доступен, достаточно выбрать пункт Проверить статус.

Перед тем как завершить работу с редактором закладок, следует сохранить внесенные изменения. Для этого необходимо выбрать пункт меню Файл\Сохранить или использовать сочетание клавиш <Ctrl+S>.

Сохранение настроек и профилей

Общие параметры. При закрытии Konqueror текущие настройки отображения (такие, как Режим просмотра, Использовать index.html и Показывать скрытые файлы в меню Вид) автоматически сохраняются и будут продолжать использоваться при запуске Konqueror в следующий раз.

Однако также имеется возможность указать для какого-либо каталога отдельные настройки. Для этого следует установить

флажок напротив пункта меню Настройка\Сохранять настройки просмотра для каталога, изменить нужные настройки и снять этот флажок. При этом в каталоге будет создан файл `.directory`, в котором будут сохранены настройки отображения. Чтобы снова показать этот каталог с обычными настройками, следует выбрать пункт меню Настройка\Удалить свойства каталога (или просто удалить файл `.directory`).

Профиль просмотра. Конqueror может сохранить весь набор параметров, касающихся внешнего вида в профиле просмотра. С программой Конqueror уже поставляются некоторые общие профили, такие, как Просмотр Web и Управление файлами, однако всегда можно добавить пользовательский.

Чтобы изменить профиль просмотра (например, Просмотр Web), следует загрузить его через пункты меню Настройка\Загрузить профиль просмотра\Просмотр Web и изменить необходимые параметры. Затем через Настройка\Сохранить профиль «Просмотр Web» произвести желаемые настройки.

Часто бывает удобно поместить на рабочий стол значок, с помощью которого Конqueror запускался бы с автоматической загрузкой определенного профиля. Для этого сначала необходимо перетащить мышью значок Конqueror из главного меню К на рабочий стол и в появившемся меню выбрать пункт Копировать сюда. Затем щелкнуть по этому значку правой кнопкой мыши, а в появившемся меню выбрать пункт Свойства..., щелкнуть по вкладке Программа и в поле Название программы ввести строку `kfmclient openProfile МойПрофиль` (вместо `МойПрофиль` следует подставить название нужного профиля). Далее во вкладке Общие надо указать новое имя и, если это необходимо, выбрать новый значок.

Модули Netscape. Конqueror поддерживает дополнительные модули Netscape. Выбрав пункт меню Настройка\Настройка Konqueror...\Модули, можно вызвать окно с двумя вкладками — Поиск и Плагины:

- во вкладке Поиск можно указать, как Конqueror должен искать новые модули — автоматически при запуске KDE или по команде (нажатие кнопки Искать новые плагины). Поиск новых модулей Конqueror проводит во всех каталогах, перечисленных в списке Сканировать каталоги;
- во вкладке Плагины перечислены все найденные модули Netscape, а также соответствующие каждому из них типы MIME и расширения файлов. Установив режим Включить

платины глобально, можно также использовать модули, содержащиеся в страницах HTML.

Модули Konqueror. В пакете `kdeaddons` содержатся дополнительные модули, которые расширяют возможности Konqueror. В табл. 4.16 приведено их краткое описание.

Таблица 4.16. Дополнительные модули Konqueror

Модуль	Функции	Вызов модуля
Перевод Web-страниц	Модуль позволяет перевести Web-страницу с одного языка на другой с помощью сервиса AltaVista BabelFish	Выбрать пункт меню Сервис \ Перевод Web-страницы
Фильтр файлов	Указать, файлы какого типа следует отображать при просмотре каталогов	Для настройки модуля надо выбрать пункт меню Сервис \ Фильтр файлов
Просмотр дерева DOM	Просмотреть список объектов текущей Web-страницы (в форме дерева)	Выбрать пункт меню Сервис \ Показать дерево DOM
Проверка HTML	Проверить текущую страницу на соответствие стандартам HTML	Сервис \ Проверка Web-страницы \ Проверка HTML
Проверка CSS	Проверить, соответствует ли стандартам описание стилей (CSS) данной страницы	Пункт меню Сервис \ Проверка Web-страницы \ Проверка CSS
Параметры HTML	Изменить некоторые параметры отображения HTML, не используя окно Настройка	Выбрать пункт меню Сервис \ Параметры HTML
Галерея рисунков	Создать Web-страницу, содержащую миниатюры всех рисунков в текущем каталоге. По умолчанию страница называется <code>images.html</code> , а миниатюры располагаются в каталоге <code>thumbs</code>	Чтобы воспользоваться этой возможностью, следует выбрать пункт меню Сервис \ Создать галерею рисунков
Изменение идентификации обозревателя	Изменять идентификацию обозревателя, не используя окно Настройка	Пункт меню Сервис \ Изменить идентификацию обозревателя
Сохранение страницы вместе с рисунками	Сохранить страницу со всеми рисунками, которые на ней имеются. Страница будет сохранена в архив, имеющий расширение <code>.war</code> . После этого ее можно открыть, щелкнув по архиву левой кнопкой мыши	Выбрать пункт меню Сервис \ Сохранить в архив
Панель проигрывателя	Проигрывать аудио- и видеофайлы во вкладке на панели навигации	Чтобы проиграть файл, достаточно перетащить его мышью на вкладку проигрывателя

4.7. Некоторые другие приложения, расширяющие функции операционных систем

В заключение будут вкратце рассмотрены программные средства, которые, расширяя функции ОС или способствуя их более эффективному выполнению, могут быть зачислены в «среды» или «оболочки», хотя и не столь безоговорочно. Это следующие продукты:

- «трехмерные менеджеры» рабочего стола MS Windows (DeskSpace, DeskLoops);
- эмулятор DOS (DOSBox);
- эмулятор ПК — MS Virtual PC;
- среда выполнения приложений MS DOS и Windows в Linux (Wine).

Трехмерные менеджеры рабочего стола



DeskSpace

DeskSpace позволяет расширить пространство на рабочем столе компьютера, размещая его на гранях куба. После инсталляции программы вместо одного рабочего стола пользователь может обозреть сразу шесть. Этими рабочими столами являются грани виртуального куба, и каждый из рабочих столов является полноценным — на нем можно открывать окна, создавать ярлыки, менять обои и т. д. Программа позволяет перемещать открытые окна между рабочими столами «перетаскиванием» их на нужную грань куба. DeskSpace поддерживает горячие клавиши для переключения между рабочими столами, а также некоторые дополнительные эффекты, например прозрачность окон.

Для активация DeskSpace необходимо нажать и удерживать комбинацию «горячих клавиш» <Ctrl+Alt+Shift>, и тогда текущий рабочий стол «погрузится» внутрь экрана, перевернется и займет место на одной из граней куба (рис. 4.30).

Для перехода к другому столу следует продолжать удерживать клавиши <Ctrl+Alt+Shift>, а затем нажать и перемещать (тащить) указатель мыши для осуществления вращения куба. Альтернативно можно использовать стрелки управления курсором <←>, <↑>, <→>, <↓> для поворота куба к необходимому столу или использовать колесико мыши.

После перехода к необходимому столу он фиксируется, если освободить клавиши <Ctrl+Alt+Shift>. DeskSpace может быть



Рис. 4.30. Пример экрана DeskSpace

настроен на использование других комбинаций активационных клавиш (отличных от $\langle \text{Ctrl}+\text{Alt}+\text{Shift} \rangle$). Кроме того, перейти к необходимому столу можно по его номеру, нажав комбинацию клавиш $\langle \text{Ctrl}+\text{Alt}+\text{N} \rangle$, где N — соответствующий номер.

Взаимосвязь между столами может поддерживаться, например, посредством «буксировки» окна с одного рабочего стола на другой. Для этого необходимо «захватить» указателем мыши заголовок окна, а затем буксировать его к краю рабочего стола. Как только указатель мыши коснется края стола, тот «погрузится» в экран и отобразится на грани куба, а затем куб повернется к пользователю смежной гранью, на которой располагается «рабочий стол назначения».

При конфигурировании DeskSpace можно выполнить широкий спектр желаемых настроек, например, для управления показом ярлыков на каждом столе следует щелкнуть правой кнопкой мыши по пиктограмме DeskSpace в линейке задач и выбрать пункт меню Manage Desktop Icons (рис. 4.31).

Чтобы DeskSpace мог отображать различные ярлыки на каждом рабочем столе, следует установить флажок Enable Different Icons On Each Desktop (Disables Auto Arrange And Auto Align). Конфигурируемый стол может быть выбран

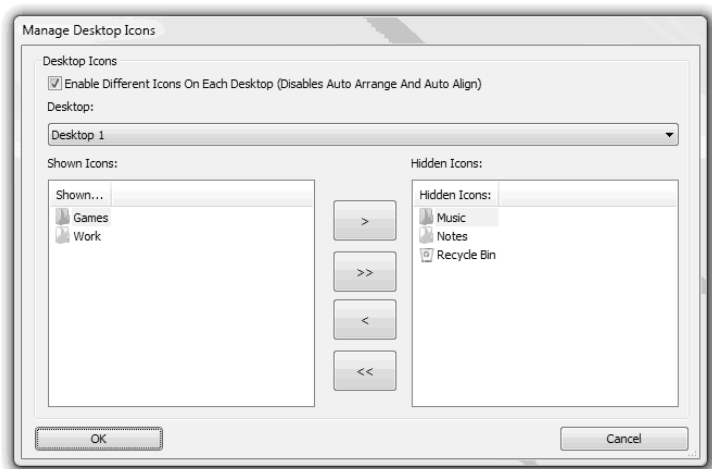


Рис. 4.31. Диалог Manage Desktop Icons конфигурирования DeskSpace

из списка Desktop, те ярлыки, которые будут отображены, представлены в списке Shown Icons, тогда как скрывающиеся — в списке Hidden Icons.

Чтобы скрыть пиктограммы этого стола, их следует выбрать в списке Shown Icons и затем нажать кнопку , а чтобы скрыть все — кнопку . Наоборот, для показа выбранных ярлыков следует использовать кнопку  для списка Hidden Icons, а для показа всех — .



Deskloops — программное средство, позволяющее организовать открытые на рабочем столе окна и управлять ими. Хотя оно и проходит под рубрикой «трехмерный менеджер рабочего стола», фактически это переключатель открытых окон. При запуске Deskloops окна автоматически организуются в подобие кольца (петли, Loop), которое может вращаться в любом направлении, по очереди помещая их в центр экрана (рис. 4.32, 3). Движение окон может активироваться либо путем помещения курсора мыши на соответствующий край экрана, либо вращением колеса мыши. С помощью курсора можно контролировать скорость вращения: если опустить его в нижнюю часть экрана, петля станет двигаться медленнее, если же поднять вверх, движение ускоряется. При этом остаются в силе такие стандартные методы переключения окон, как панель задач и клавиши <Alt+Tab> (рис. 4.32, 2).

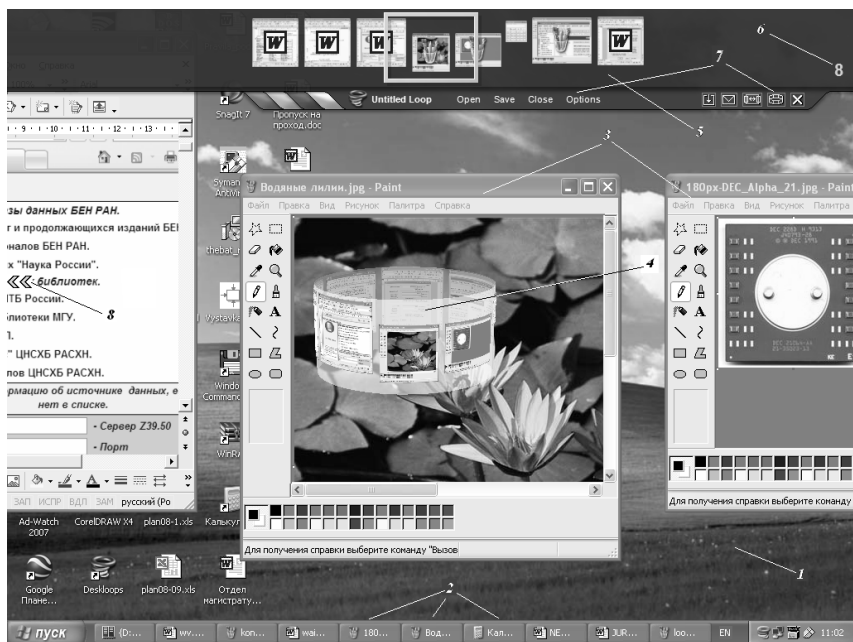


Рис. 4.32. Вид рабочего стола под управлением Deskloops:

1 – рабочий стол; 2 – линейка задач, на которой отмечены открытые окна; 3 – окна приложений в натуральную величину на столе; 4 – изображение петли (Loop) окон; 5 – лента (Strip) окон; 6 – счетчик числа открытых окон; 7 – кнопки управления лентой; 8 – указатель направления вращения петли

«Петля» подобна эластичной ленте, которая «сжимается» при закрытии окна и, наоборот, «растягивается» при открытии. Наглядный образ этого процесса помещается в центр экрана (рис. 4.32, 4), чтобы пользователю было легче представить себе, где именно за пределами видимой области стола расположено то или иное окно, на экране отображается небольшая трехмерная копия петли. Она возникает на экране всякий раз, когда пользователь вращает ленту из окон.

Если открыто много окон, переключаться между ними с помощью петли не всегда удобно. Поэтому в Deskloops предусмотрен еще один инструмент для этого — лента (Strip, рис. 4.32, 5), которая размещается у верхнего края экрана и активируется при подводе курсора. На ней представлены эскизы всех окон в том же порядке, в котором расположены на петле, и для перехода к нужному окну достаточно щелкнуть правой кнопкой по его эски-

зу. Если же эскизы не помещаются на экране, ленту можно прокрутить, используя горизонтальный скроллинг. Лента содержит также счетчик числа открытых окон и кнопки управления, с помощью которых можно:

 — максимизировать размер всех окон (maximize all windows at once);

 — организовать окна в петлю с равными интервалами (arrange);

 — сохранить петлю (quicksave) и т. п.

Эмуляторы DOS

 Рассматриваемое здесь приложение DOSBox является одним из представителей программных средств, позволяющих воспроизвести среду исполнения DOS-приложений на другой платформе. Известны также эмуляторы для Unix/Linux — Dosemu или VMWare, система управления виртуальными компьютерами для Windows и Linux, которая, помимо среды выполнения для запуска ОС Windows, Linux, NetWare и FreeBSD, способна создавать виртуальную среду для запуска DOS.

DOS, загружаемая в DosEmu и VMWare, является почти произвольной системой, никак не зависящей от типа внешней ОС. Внутри DosEmu может быть запущена и MS DOS, и PC DOS, и Novell DOS, и распространяемая вместе с DosEmu FreeDOS, и даже Windows 3.1.

Эмуляция представляет собой покомандную программную интерпретацию всего подлежащего выполнению кода, т. е. программное моделирование центрального процессора и всего остального оборудования. Как правило, эмуляция широко применяется при разработке нового аппаратного обеспечения, а также при разработке ПО для небольших устройств, которые не приспособлены для загрузки отладчика. В последнем случае отлаживаемая программа запускается внутри эмулятора устройства, который сам, в свою очередь, является ядром отладчика.

Несмотря на медлительность, у программных эмуляторов есть несколько достоинств:

- эмулятор способен работать на любой платформе, а не только на той, которая имеет аппаратную совместимость с запускаемой программой;

- эмулятор позволяет создать запускаемой программе такую среду выполнения, которую программа будет не в состоянии отличить от настоящего ПК;
- ход выполнения программы теоретически может быть контролирован с максимально возможной точностью;
- исходные тексты эмулятора обычно открыты для всех желающих и поддаются исправлению.

DOSBox эмулирует не IBM PC вообще, а IBM PC, на котором запущена DOS. Это означает, что вместо загрузки операционной системы откуда-либо DOSBox сам помещает в ОЗУ эмулируемого компьютера все необходимые данные и исполняемый код.

DOSBox версии 7 существует в вариантах для Windows, BeOS, Linux, MacOS X и других ОС. DOSBox эмулирует процессоры i286/386 в реальном/защищенном режимах, файловую систему с директориями /XMS/EMS, графические адаптеры Tandy/Hercules/CGA/EGA/VGA/VESA, карты SoundBlaster/Gravis Ultra Sound для вывода звука старых приложений.

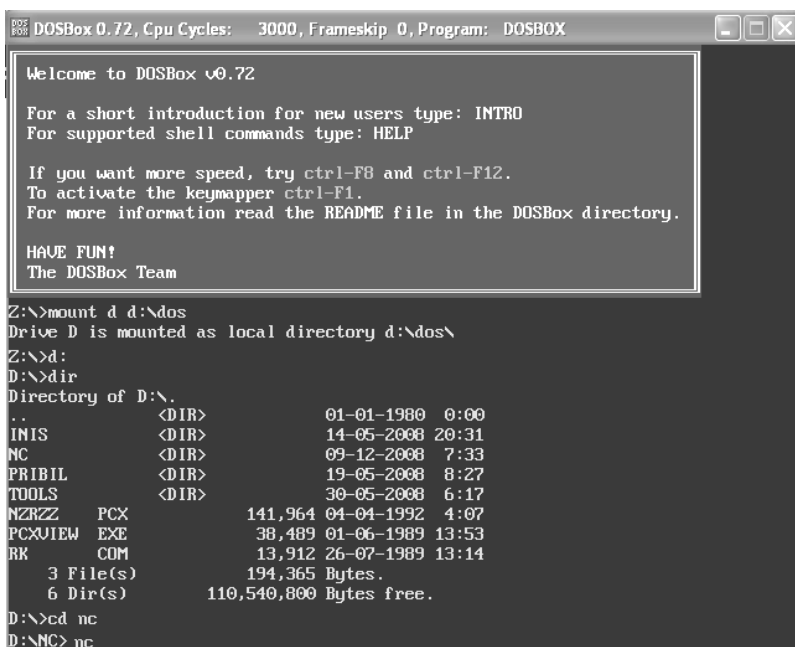


Рис. 4.33. Стартовый экран DOSBox

На рис. 4.33 приводится вид стартового экрана DOSBox. После запуска приложения были выполнены команды:

- монтирования каталога `d:\dos` как дисковод `d:`;
- перехода на диск `d:`;
- просмотра оглавления текущего каталога (`d:`);
- перехода в подкаталог `d:\nc`;
- запуска NC.

После запуска Norton Commander экран приобретает вид, приведенный на рис. 2.1.

Microsoft Virtual PC 2007



Virtual PC позволяет создать одну или более виртуальных машин (гостевые системы), каждая из которых использует свою операционную систему, на одном физическом компьютере (хост-система). Виртуальная машина (ВМ) эмулирует стандартный x86-совместимый ПК, включая все основные аппаратные компоненты, за исключением процессора. Используя эмулируемые аппаратные средства и процессор физического ПК, каждая ВМ работает как отдельный физический компьютер (рис. 4.34).

Продукт Virtual PC (VPC) был приобретен корпорацией Microsoft вместе с компанией Connectix и впервые под торговой маркой Microsoft был выпущен как Microsoft VPC 2004. За прошедшее время Microsoft смогла значительно усовершенствовать приложение, добавив новые возможности, и в конце 2006 г. вышла первая версия VPC 2007. Продукт VPC предназначен для за-

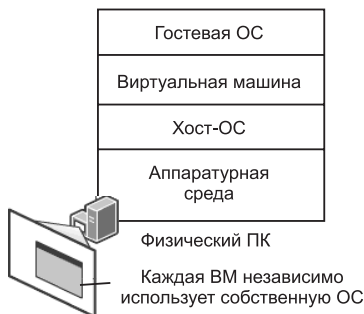


Рис. 4.34. Взаимодействие «слоев» программных и технических средств в VPC 2007

пуска одной или нескольких гостевых операционных систем на настольных системах, прост в использовании и ориентирован на рядового пользователя.

Перечислим основные возможности новой версии продукта VPC:

- оптимизация платформы под Windows Vista. В VPC 2004 также можно было установить Windows Vista, однако в новой версии продукта система работает быстрее и стабильней;
- увеличение быстродействия за счет использования усовершенствований, введенных в серверной платформе виртуализации Microsoft Virtual Server 2005;
- поддержка 64-битовых хостовых операционных систем Windows — на 64-битовой Windows Vista можно установить гостевую систему любой разрядности;
- поддержка аудиоустройств в гостевых системах Windows Vista.

Microsoft рекомендует использовать программу VPC 2007 в следующих целях:

- запуск более старых операционных систем в виртуальных машинах для обеспечения совместимости программ. Некоторые новые возможности и изменения Windows Vista, такие как Windows Resource Protection (WRP), могут повлиять на эффективность работы программ, предназначенных для более ранних версий Windows;
- разработка и тестирование приложений;
- техническая поддержка — можно установить несколько экземпляров операционных систем для того, чтобы персонал службы технической поддержки, запуская нужную конфигурацию операционной системы, мог обеспечивать разрешение проблем пользователей в максимально сжатые сроки;
- контроль качества программного обеспечения и запуск приложений в изолированной виртуальной среде, где не опасно повреждение системы;
- обучение пользованию различными операционными системами и приложениями без необходимости приобретения дополнительных компьютеров.

На рис. 4.35 приведен пример ситуации, когда гостевая система (Windows 2000) запущена на виртуальной машине VM_LIN2000 в среде Windows XP. Здесь же можно видеть окно приложения Far Manager, выполняющегося в гостевой ОС, и окно консоли виртуальных машин.



Рис. 4.35. Запуск виртуальной машины с Win2000 в среде WinXP:

1 — рабочий стол хост-системы (Windows XP); 2 — окно ВМ VM_LIN2000; 3 — меню Action окна ВМ; 4 — окно приложения Far, запущенного на ВМ; 5 — окно консоли ВМ

Далее, на рис. 4.36 можно видеть, как гостевая система Linux Suse 11.0 (интерфейс Gnome) запущена на той же виртуальной машине VM_LIN2000 в среде Windows XP. На экране здесь размещается консоль ВМ, а также окно ее отключения (Close\ Turn off).

Структура и состав ВМ. По умолчанию все файлы, образующие ВМ, собраны в одном каталоге для обеспечения простоты настройки и переносимости. Эти файлы включают:

- файл конфигурации ВМ (.vmc) в XML-формате, который содержит конфигурационную информацию ВМ, включая все настройки и определения эмулируемых аппаратных средств, созданные Мастером ВМ (New Virtual Machine Wizard);
- файл виртуального жесткого диска (.vhd), созданного Мастером ВМ или Мастером НЖМД (Virtual Disk Wizard — рис. 4.37, в);

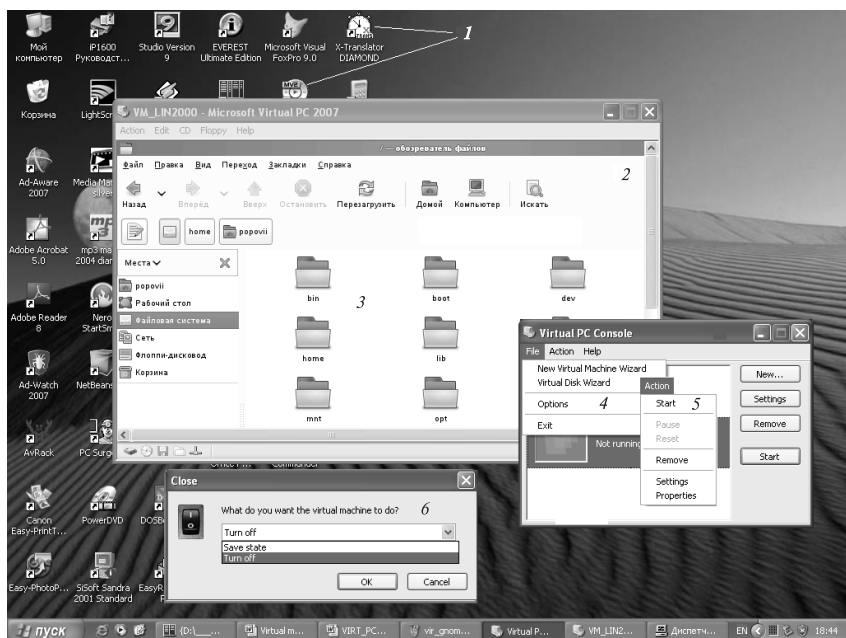


Рис. 4.36. Запуск виртуальной машины с Linux в среде WinXP:

1 — рабочий стол хост-системы (Windows XP); 2 — окно VM VM_LIN2000 (просмотр корневого каталога FC Linux); 3 — окно консоли VM; 4 — меню File; 5 — меню Action; 6 — окно отключения VM (Turn off)

- при необходимости ряд других файлов, например, файл сохраненного состояния VM (.vsv), используемый при рестарте VM.

По умолчанию каталоги VM размещаются по пути My Documents\My Virtual Machines, а все перечисленные файлы располагаются в каталоге с именем, повторяющим имя VM.

Консоль VM (Virtual PC Console) является исходной точкой старта и пунктом управления совокупностью установленных VM и предусматривает следующие возможности:

- позволяет вызвать Мастера новых VM и Мастера виртуальных дисков;
- показывает список установленных VM и их состояние;
- позволяет осуществить настройку конфигураций VM, включая как глобальные опции, прилагаемые ко всем установленным VM, так и индивидуальные настройки каждой VM.

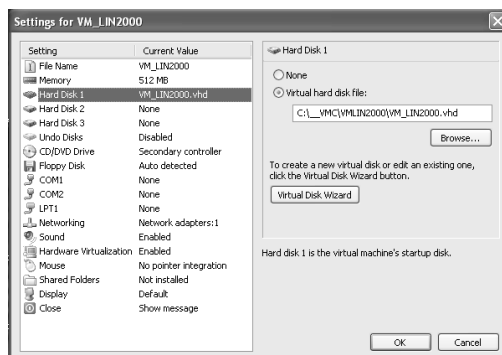
Консоль осуществляет доступ и управление виртуальными машинами посредством меню **File** и **Action** (табл. 4.17).

Таблица 4.17. Некоторые функции консоли виртуального ПК

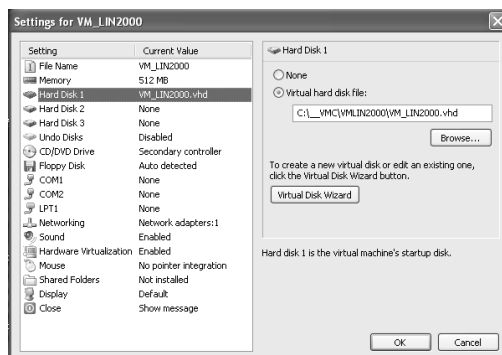
Функция VM	Действия	Вызов функции
Настройка Settings	Открывается диалоговый экран Settings (рис. 4.37, а)	Кнопка Setting или меню Action\Setting
Пуск Start	Запуск виртуальной машины, а также операционной системы, если она уже установлена (в противном случае появляется приглашение указать загрузочное устройство)	Кнопка Start или меню Action\Start
Временный останов Pause	Приостанавливает работу запущенной VM	Action\Pause или из меню окна VM (рис. 4.37, 3)
Возобновить Resume	Возобновляет работу приостановленной VM	Если выбранная машина приостановлена — из меню Action\Resume
Свойства Propeties	Показывает диалоговое окно свойств (рис. 4.37, б), содержащее информацию об ОС, процессоре, памяти, загрузке контроллеров НЖМД, видеокарты и пр.	File\Properties
Выключить Close	Останавливает и выключает выбранную VM. Могут выполняться действия, описанные в пункте Close диалога Setting (табл. 4.18)	Кнопка Close или меню File\Exit
Сброс/рестарт Reset	Рестарт VM без выключения	Action\Reset
Удалить Remove	Удаление выбранной VM из списка зарегистрированных	Кнопка Remove или меню Action\Remove

Дополнения VM (Virtual Machine Additions). Установка Virtual Machine Additions осуществляется через меню Action\Install or Update Virtual Machine Additions (или <Правый Alt+I>) — см. рис. 4.35, 3. Эта возможность позволяет не только увеличить быстродействие гостевой системы, но сделает доступными следующие функции:

- возможность перетаскивания (Drag&Drop) файлов и папок из хост-системы в гостевую и обратно;
- при изменении размеров окна консоли гостевой системы разрешение внутри нее будет автоматически подгоняться под размер окна;



а



б



в

Рис. 4.37. Экран настроек (Setting) виртуальной машины (а); окно Properties (б); Мастер жестких дисков (в)

- улучшенный драйвер виртуального видеоадаптера позволит установить произвольное разрешение и глубину цвета;
- фокус мыши будет автоматически передаваться в гостевую систему при попадании курсора в пределы окна консоли и возвращаться в хост-систему при уходе указателя из этой зоны. Это избавляет от необходимости нажимать <Правый Alt> для того, чтобы передать фокус в гостевую систему, и позволяет работать с окном консоли гостевой системы как с обычным окном приложения;
- обмен файлами с хост-системой будет возможен также через общие папки (Shared Folders);
- общий буфер обмена у гостевой и хост-систем, что упрощает их взаимодействие

Настройка виртуальной машины. Каждая ВМ имеет настройки, которые определяют различные функции и свойства. Некоторые из настроек доступны при выполнении New Virtual Machine Wizard, и все доступны из консоли ВМ (рис. 4.37). Основные настройки поясняются в табл. 4.18.

Таблица 4.18. Параметры настроек виртуальных машин

Параметр	Описание
File Name	Здесь можно изменить имя файла настроек ВМ (соответственно изменится и имя самой машины)
Memory	Указание количества физической памяти хоста, выделяемой гостевой системе
Hard Disk 1 – Hard Disk 3	Может быть изменен уже имеющийся или добавлен новый виртуальный жесткий диск. При нажатии кнопки Virtual Disk Wizard откроется мастер создания и изменения дисков (рис. 4.37, в)
Undo Disks	Если поставить флажок этого пункта, то все изменения, которые произошли на дисках во время работы с ВМ, в конце сеанса работы с ней следует либо подтвердить, либо отменить. Включение такой функции потребует наличия дополнительного места на физическом жестком диске
CD/DVD Drive	Позволяет монтировать физические CD- и DVD-приводы в виртуальной машине (следует установить флажки)
Floppy Disk	Включение автообнаружения флоппи-дисков
COM1–COM2, LPT1	Подключение портов COM и LPT к ВМ
Networking	Настройка сетевых адаптеров, доступных для ВМ

Окончание табл. 4.18

Параметр	Описание
Sound	Позволяет проигрывать звук в виртуальной машине, используя физический аудиоконтроллер хоста
Mouse	Интеграция указателя мыши с гостевой системой. Такая возможность появляется с установкой Virtual Machine Additions
Shared Folders	Создание общих папок для обмена файлами между гостевой и хостовой ОС. После создания папка будет доступна как сетевой диск в гостевой системе. Опция доступна только при установленных Virtual Machine Additions
Display	Выбор поведения экранных настроек гостевой системы и, в частности, возможность старта в полноэкранном режиме, настройки разрешения и окна консоли гостевой системы. Комбинация клавиш <Alt+Enter> во время работы с гостевой системой, позволяет переходить в полноэкранный режим и обратно
Close	Здесь можно выбрать действие, автоматически совершаемое при закрытии гостевой системы, а также отредактировать меню, возникающее при выборе пункта Close (см. рис. 4.36, б)

Мастер жестких дисков Virtual Disk Wizard. При создании нового диска этот мастер позволяет выбрать один из четырех типов дисков (см. рис. 4.37, в):

- *Dynamically expanding* (по умолчанию) — динамически расширяющийся тип диска, который автоматически увеличивает свой размер при записи данных на него, однако для сжатия диска при высвобождении места на нем потребуется использовать этот мастер;
- *Fixed size* — свободное место для диска резервируется при его создании, поэтому файл виртуального диска имеет постоянный объем. Такие диски работают немного быстрее динамически расширяющихся;
- *Differencing* предназначен для хранения изменений уже существующего диска и связан с ним;
- *Linked to a hard disk.* При использовании такого диска ВМ будет непосредственно записывать данные на физический диск.

Всего можно подключить к виртуальной машине до двух дополнительных дисков помимо основного. После создания диска в гостевой системе он должен быть отформатирован в нужной файловой системе.

Wine



Wine — свободное программное обеспечение, позволяющее пользователям UNIX-подобных систем архитектуры x86 (и других архитектур при наличии совместимости) выполнять приложения Microsoft Windows. Кроме того, Wine предоставляет инструментальный разработчики программ для переноса унаследованных исходных кодов из среды Windows в среду UNIX путем простой перекомпиляции.

На рис. 4.38 приводится экран, иллюстрирующий выполнение Windows-приложения notepad в среде Wine.

«Wine» является рекурсивной аббревиатурой и расшифровывается как «*Wine Is Not an Emulator*» («Wine — это не эмулятор», имеется в виду, что Wine не является эмулятором компьютера, как, например, Qemu или VirtualBox).

Первоначально Wine разрабатывалась как средство запуска 16-битовых приложений Windows 3.1 в системе GNU/Linux. Со временем Wine была портирована на другие UNIX-подобные системы, такие как FreeBSD, Solaris и т. д., а также на Windows

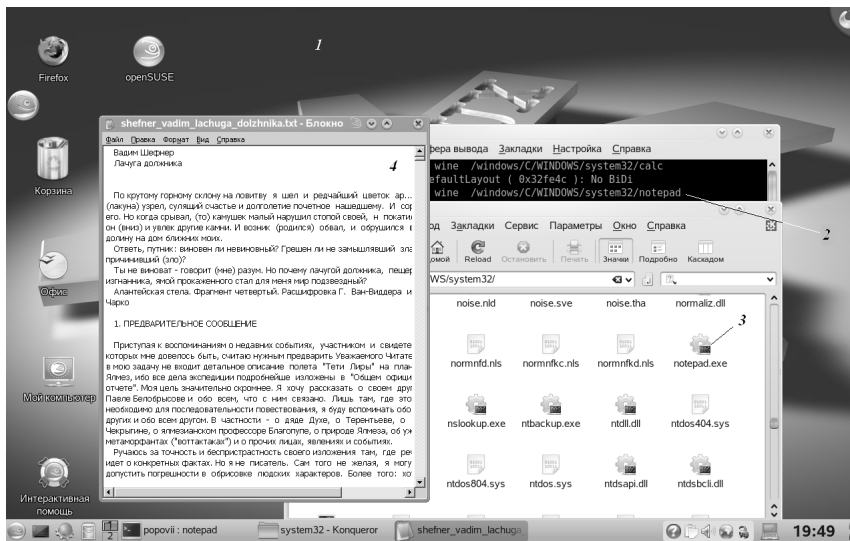


Рис. 4.38. Иллюстрация к выполнению приложения Windows в среде Wine:
1 — рабочий стол KDE Suse Linux; 2 — строка запуска Wine в окне Терминал;
3 — ярлык Win-приложения notepad.exe (аксессуар Блокнот); 4 — окно приложения

(для запуска старых приложений на новых версиях ОС). Была добавлена поддержка 32-битовых приложений Windows, а затем и 64-битовых.

Wine воспринимает системные вызовы Windows-приложений к библиотекам операционной системы и замещает их своими. Таким образом, эмуляция команд процессора не происходит, и приложения выполняются в Wine почти так же быстро, как и в исходной операционной системе. Для своей работы Wine не требует наличия установленной ОС Windows, хотя и может использовать ее библиотеки.

Код проекта Wine использует ряд коммерческих и свободных продуктов, в том числе:

- Crossover Office — поддерживает использование приложений MS Office и Adobe PhotoShop на Linux;
- Cedega — может быть использована для выполнения Windows-игр на Linux;
- WineTools — управляемый через меню инсталлятор для более чем 100 Windows-приложений, разработанных для платформы x86 (AMD Athlon или Intel Pentium) в среду Linux с использованием Wine;
- IES4Linux — утилита, созданная специально для использования Internet Explorer в Wine. Обеспечивает легкий автоматический процесс установки и поддержки Internet Explorer версий 5 и 6.

Контрольные вопросы

1. Что такое программные оболочки и какие функции они выполняют?
2. Какие вы знаете оболочки для DOS и для Windows? Что общего между ними?
3. Какие оболочки относятся к «нортонообразным», а какие нет?
4. Какие функции выполняет Windows Commander?
5. Какие функции выполняет Norton Commander для Windows?
6. Как создать каталог и файл в Norton Commander?
7. Укажите назначение основных функциональных клавиш NC.
8. Как убрать (восстановить) панели в Norton Commander?
9. От чего зависит действие Norton Commander при нажатии <Enter>?
10. Зачем служит файл `nc.ext` и какова его структура?
11. Зачем служит файл `nc.mnu` и какова его структура?
12. Какие пункты управляющего меню есть у Norton Commander?

13. Какие дисковые функции имеются у Norton Commander?
14. Как изменить атрибут файла (группы файлов)?
15. Как сделать видимыми (или невидимыми) скрытые и системные файлы?
16. Что такое программа PCTools и какие функции она выполняет?
17. Как запустить программу PCTools и как из нее выйти?
18. Какие меню имеет программа PCTools и чем они отличаются?
19. Просмотрите дерево каталогов с помощью PCTools и скопируйте группу файлов в каталог `primer`.
20. Просмотрите план диска и карту дискеты с помощью PCTools.
21. Что такое программа Dosshell и какие функции она выполняет?
22. Как запустить программу Dosshell и как из нее выйти?
23. Чем программы Dosshell и PCTools отличаются друг от друга?
24. Каковы основные элементы окна обозревателя Konqueror?
25. Каковы функции средней кнопки мыши при работе с Konqueror?
26. Что такое Панель навигации? Перечислите вкладки панели и поясните их назначение.
27. Каковы возможности Konqueror в режиме Web-обозревателя?
28. В чем заключается настройка Konqueror?

Задания

1. Уберите, а затем восстановите панели НС и поменяйте их местами.
2. Создайте каталог `test` и в нем создайте текстовый файл `goga.txt`. Затем удалите этот файл и каталог.
3. Произведите сортировку файлов по имени, расширению, дате создания и размеру в каталоге `windows`.
4. Найдите на диске файл `nc.exe`.
5. Найдите на диске каталоги `dos`, `windows` и `system`. Выделите в них один файл, группу файлов, все файлы, а затем отмените выделение.
6. Вызовите помощь, затем меню пользователя и управляющее меню НС.
7. Выведите дерево каталогов диска `c:` и перейдите в каталог НС.
8. Получите всю информацию о диске и дискете. Определите, сколько свободного места осталось на диске и дискете.
9. Отредактируйте файл `nc.ext` и свяжите файлы с соответствующими приложениями. Проверьте работу нового файла `nc.ext`.
10. Создайте локальное меню и отредактируйте файл `nc.mnu`. Запустите программы из этого меню.
11. Отформатируйте дискету, создайте метку на дискете, протестируйте дискету с помощью PCTools.

12. Восстановите стертый файл с помощью PCTools.
13. Произведите поиск файла `win.com` на диске с помощью PCTools.
14. Скопируйте группу файлов с диска на дискету, просмотрите их, а затем удалите с помощью PCTools.
15. Создайте и отредактируйте текстовый файл с помощью программы PCTools.
16. Запустите Far, произведите копирование файлов и каталогов и архивацию файлов.
17. Запустите Windows Commander, произведите копирование файлов и каталогов и распаковку архива.
18. Запустите Norton Commander для Windows и запустите основные приложения Windows.
19. Запустите Far Manager и осуществите основные операции с файлами и каталогами (копирование, сортировка, переименование, перенос, удаление), запустите прикладные программы на исполнение.
20. Просмотрите дерево каталогов с помощью программы Dosshell и скопируйте группу файлов в каталог `test_primer`.
21. Просмотрите карту диска, карту файлов и карту дискеты с помощью программы Dosshell.
22. Отформатируйте дискету, создайте метку на дискете, протестируйте дискету с помощью программы Dosshell.
23. Запустите на исполнение любую программу с помощью программы Dosshell.
24. Произведите поиск файла `win.com` на диске с помощью программы Dosshell.
25. Скопируйте группу файлов с диска на дискету, просмотрите их, а затем удалите с помощью программы Dosshell.
26. Создайте и отредактируйте текстовый файл с помощью программы Dosshell.
27. Запустите Konqueror в режиме менеджера файлов, действиями Копировать и Вставить перенесите группу файлов из каталога `/usr` на дискету.
28. Просмотрите оглавление дискеты, удалите один из файлов.
29. Добавьте к интерфейсу Konqueror панель терминала. Последовательностью команд `cd` перейдите на дискету, просмотрите оглавление командой `ls`.
30. Разделите экран главного окна Konqueror. Поместите на левую панель дискету, на правую — каталог `/home`. Скопируйте группу файлов с дискеты на жесткий диск.
31. Запустите Konqueror в режиме Web-обозревателя. Введите адрес `www.inion.ru`. Просмотрите заглавную страницу сайта, сделайте копию и сохраните на дискете.

Заключение

Операционные системы ЭВМ развиваются и модифицируются в общем контексте развития технических и программных средств. Постоянной средой этого развития является сосуществование по меньшей мере трех уровней организации информационно-вычислительных процессов — *аппаратурного, программного, информационного*. Они образуют некоторые *слои, страты* информационных технологий, которые взаимозаменяемы в определенных пределах, например функция $y = e^x$ может быть вычислена:

1 — *аппаратурно* (усилитель с нелинейными обратными связями);

2 — *программно* (вычисление суммы ряда для e^x);

3 — на основе *базы данных*, в качестве таковой здесь может фигурировать сборник математических таблиц, содержащих соответствующие значения y и x .

В рамках программного обеспечения следует в свою очередь упомянуть известные *подслои* — *операционные системы, средства разработки приложений, собственно приложения*.

Необходимо отметить то не всегда очевидное обстоятельство, что перечисленные слои технических и программных средств сложились в результате длительной (по масштабам информатики!) эволюции*. Они приспособились друг к другу и взаимодействуют так же, как живые организмы в земной биосфере. Есть растения, которые связывают углекислый газ и выдают кислород, а также производят жиры, крахмал, клетчатку, есть травояд-

* *Washington Post*, 1984 г.: «В 1953 г. ЭВМ с памятью 64 К стоила 1 млн долл., сейчас она стоит менее 1 тыс. долл. Если бы автомобили развивались в течение поледних 20 лет теми же темпами, как компьютеры, то сегодня Роллс-Ройс стоил бы \$3.0, проходил бы миллион миль на галлоне бензина, развивал бы мощность лайнера Queen Elisabeth, и 2 автомобиля помещались бы на кончике пера».

ные, которые едят растения, есть хищники, которые едят травоядных, есть человек (венец творения), который ест всех. Но бесперспективны попытки заставить человека усваивать «продукт вторичный» (см. Вл. Войнович, «Москва, 2042»), и также могут оказаться бесперспективными работы, разрушающие указанную структуру и объединяющие функции различных слоев, хотя они постоянно идут и их результаты даже иногда попадают на рынок (программных средств).

Если не учитывать вероятность «технологических революций» (полный отказ от фон-неймановских машин, например), то основные направления развития информационных технологий следует ожидать в «диффузии» процессов обработки информации между различными слоями (аппаратурный, программный, информационный) и подслоями программного слоя (операционная система, СУБД и пр.). Самый расхожий пример (ему лет 40) — *макрокоманды* и *микропрограммы*. В первом случае речь идет о программной эмуляции команд процессора (иногда эмуляция системы команд одной машины на другой), во втором — «зашивке программ в железо». Более поздние примеры — эмуляция терминалов, эмуляция протоколов модемов и пр.

Следующий пример (в подслоях) — операционные (ОС) и информационные системы (ИС). Обычные ОС (OS/360, RSX, MS DOS, Win/95/98/NT) достаточно эффективно обеспечивают *доставку файла* прикладной программе (осуществляют *связь имени с адресом* и, не интересуясь содержимым, возлагают эту проблему на пользователя). ИС (СУБД, АИПС) реализует *доставку записи (или строки) файла* в прикладную программу (осуществляет *связь содержания с адресом*).

Тем не менее известны ОС, включающие элементы ИС, например ОС Unix включает команды `sort` (сортировка текстов), `grep` (контекстный поиск в файле), `awk` (генерация отчетов) и пр. (соединение отношений, например). Достаточно распространенная ОС PICK содержит язык запросов (ЯЗ) ENGLISH, возможны и другие примеры (например, индексные методы доступа в OS/360).

Однако это не более чем внешнее сходство, поскольку ИС обеспечивают выполнение всех этих функций за счет вспомогательных файлов и процессов (индексирование, хэширование, динамически отслеживающих изменение данных), в то время как ОС — за счет прямого сканирования содержания (заведомо неэффективного для больших файлов).

Кроме данного *вертикального* взаимопроникновения компонентов, может быть рассмотрено горизонтальное (*клиент-серверное*), когда функции переработки информации, например, перемещаются от центрального процессора к внешним устройствам ЭВМ:

- интеллектуальный терминал (берет на себя функции предобработки и синтаксического анализа запроса пользователя);
- интеллектуальный принтер (управляется некоторым языком, начиная от ESC-последовательностей, стандарта Epson и заканчивая языками управления плоттерами и принтерами Hewlett-Packard);
- интеллектуальный видеомонитор (карты-ускорители алгоритмов обработки изображений, описанных на специализированных языках);
- интеллектуальный контроллер магнитного диска (машина баз данных, data base machine, управляется директивами find, select, sort и передает в центральный процессор только релевантную информацию). Идея, весьма популярная в 80-е годы, но не получившая в последующем распространения, сколько-нибудь заслуживающего внимания.

Данные процессы диффузии (как и вертикальные) двунаправлены — достаточно вспомнить алфавитно-цифровые принтеры и терминалы (аппаратная, даже механическая генерация символов) и графические принтеры и терминал (программная генерация). Хороший пример неудачного возложения функций процессора на терминал — протокол устройства IBM 3270 (пользователь продолжает работать, не зная, что хост-машина, например, выключилась). Возврат назад — передача ряда функций терминала (обработка ввода и вывода) центральному процессору — приводит к ANSI-терминалам, наиболее распространенным сегодня.

Рассмотренные двунаправленные горизонтальные и вертикальные перемещения элементов технологий вызывают аналогию с *океанской волной*, которая, устрашающе выглядя на поверхности, переносит потоки энергии, но частицы воды под ней описывают круговые траектории, в принципе оставаясь на месте.

Литература

1. Бек Л. Введение в системное программирование. М.: Мир, 1988.
2. Дегтярев Е. К. Введение в Unix. М.: МП «Память», 1992.
3. Дьяконов В. Ю., Китов В. А., Калинин И. А. Системное программирование: учеб. пособие для вузов / под ред. А. Л. Горелика. М.: Высш. шк., 1990.
4. Калверт Ч. Программирование в Windows 95. Освой самостоятельно. М.: Восточная Книжная Компания, 1996.
5. Керниган Б. В., Пайк Р. Unix — универсальная среда программирования. М.: Финансы и статистика, 1992.
6. Мэсфилд Р. Windows 95 для занятых, 1998.
7. Петерсен Р. Linux: руководство по операционной системе. Киев: Издательская группа BHV, 1997.
8. Петроченков А. А. Компьютер и периферия. М., 1995.
9. Попов И. И. Автоматизированные информационные системы (по областям применения): учеб. пособие / под общ. редакцией К. И. Курбакова. М.: Изд-во Рос. экон. акад., 1998.
10. Попов И. И., Храмов П. Б. Мировые информационные ресурсы и сети (методы доступа к ним): учебник / под ред. К. И. Курбакова. М.: Изд-во Рос. экон. акад., 1998.
11. Потапкин А. В. Операционная система Windows 95, 1997.
12. Робачевский А. М. Операционная система Unix. СПб.: BHV Санкт-Петербург, 1997.
13. Стинсон К. Эффективная работа в Windows 95, 1996.
14. Титаренко С. П. Управление процессами в современных операционных системах ЭВМ: учеб. пособие. Изд. Белгородской государственной технологической академии строительных материалов, 1999.
15. Фигурнов В. Э. IBM PC для пользователя. М.: Финансы и статистика, 1990.

16. *Фойц С.* Windows 3.1, 1995.
17. *Франкен Г., Молявко С.* MS DOS 6.2 для пользователя. Киев: ВНУ, 1994; М.: БИНОМ, 1994.
18. www.computerra.ru/reviews
19. www.kentos.ru
20. www.moodle.ipm.kstu.ru
21. www.opennet.ru
22. www.opensuse.org
23. www.samag.ru
24. www.XServer.ru

Глоссарий (список используемых терминов)

API (Application programming interface) — интерфейс прикладного программирования — предопределенный набор функций, которые операционная система предоставляет в распоряжение приложений. Определяет состав, параметры и смысл функций, предоставляемых ОС программисту. Например стандарт POSIX на API ОС Unix включает функции: набор файловых операций, операции со строками, функции многозадачности, управления процессами, управление терминалом. Помимо POSIX есть API Win32 и т. п. API ОС может включать самые разнообразные услуги, вплоть до поддержки функций телефонного аппарата на базе модема (TAPI Win32).

BIOS (Plug & Play BIOS) — базовая система ввода-вывода персонального компьютера. BIOS обеспечивает интерфейс самого низкого уровня с такими устройствами, как системные часы, жесткий диск и монитор. Plug & Play BIOS дополняет функции BIOS рядом процедур, поддерживающих некоторые действия подсистемы Plug & Play, например перечисление устройств.

Cat (команда Unix) выводит содержимое файла на экран.

Cd (команда Unix) устанавливает указанный каталог текущим рабочим каталогом.

Cp (команда Unix) — копирование содержимого файла в файл с другим именем либо в другой каталог с сохранением существующего имени файла, всех файлов одного каталога в другой каталог.

Демон (демон) — сокращение от Disk and execution monitor (отслеживание дисковых операций и выполнения программ). Демоны присутствуют на всех Unix-системах и выполняют задачи, не требующие вмешательства пользователя. Читатели, знакомые с Microsoft Windows, могут сравнить демонов и задачи, выполняемые ими, со «службами». В качестве примера можно привести демон LPD (Line Printer Daemon — Демон почтовой печати), присутствующий на большинстве Unix-компьютеров.

Drag-and-Drop, D-n-D («захватить-и-перетащить», «перетащить-и-отпустить», «буксировать», «перетаскивать» и пр.) — элемент технологии интерфейсов WIMP, состоящий из следующих действий — «захват» экранного объекта (ярлык, имя файла и пр.) с помощью указателя мыши,

«букировка» к месту назначения на экране при нажатой клавише, «сбрасывание» объекта при отпускании клавиши.

FAT — таблица размещения файлов.

Find (команда Unix) — поиск файлов. Параметр `-name` — поиск файлов с указанным именем. Параметр `-print` — вывод имен найденных файлов на экран, этот параметр обязателен, если пользователь хочет увидеть результат поиска.

Ghostscript — программный эмулятор растрового процессора Raster Image Processor (RIP) для PostScript, разработанный L. Peter Deutsch. Ghostscript используется в Linux- и Unix-системах для преобразования файлов в формате PostScript в растровый формат для отправки на устройства, не поддерживающие PostScript.

GNOME (GNU Network Object Model Environment) — одна из распространенных графических сред для Unix.

GNU — проект разработки операционных систем и их сред, свободно распространяемых, совместимых с Unix и переносимых, так чтобы пользователи ОС Unix могли легко перейти к ним. Название было выбрано согласно хакерским традициям, как рекурсивный акроним «GNU is Not Unix» (GNU — это вам не Unix!).

GUI (Graphical User Interface) — графический интерфейс пользователя. Всякая графическая среда (включая KDE) — это GUI. Большинство GUI включают средства для работы мышью и оконный менеджер.

KControl — центр управления KDE, поддерживающий все многообразие настроек.

KDE (K* Desktop Environment) — ведущая визуальная среда для рабочих станций Unix. Рабочий стол KDE совмещает простоту использования, гибкость настройки, уникальный графический дизайн с технологическими достоинствами операционной системы Unix.

Kicker — панель рабочего стола KDE — элемент, чаще всего располагающийся внизу экрана.

Konqueror — ключевой элемент KDE, воплощающий в себе файловый менеджер, Web-обозреватель, просмотрщик текстов и прочие возможности.

KParts — технология встраивания компонентов, благодаря которой приложения KDE могут встраивать в себя другие приложения KDE. Например, Konqueror использует как KPart — встроенный просмотрщик текстовых файлов.

Kprinter — программа печати, используемая всеми приложениями KDE, а также другими, например, gv, Acrobat Reader, Netscape, Mozilla, Galeon, StarOffice, OpenOffice и программами из GNOME. Kprinter может быть использована отдельно, если запустить ее из графической консоли или из «Mini-CLI». Может быть распечатано множество различных файлов из разных папок, в разных форматах одновременно в одном задании печати.

* Что именно означает здесь «К», читателю предоставляется выяснить самостоятельно.

Logname (команда Unix) — вывод системного идентификатора пользователя.

Ls (команда Unix). При вводе без параметров выдает список файлов и подкаталогов текущего каталога. При вводе с параметром *-i* — вывод индексов файлов.

Mkdir (команда Unix) — создание нового каталога (каталогов).

MV (команда Unix) — переименование файла или перемещение одного или нескольких файлов в другой каталог.

MV (команда Unix). Эта команда перемещает указанный файл в указанный каталог.

OLE (связывание и внедрение объектов) — реализованная в системах Windows архитектура *Component Object Model (COM)*.

Page (команда Unix) выводит на экран весь файл или его части.

PCI bus — разработанная Intel *шина*, которая предназначена для поддержки высокоскоростного 32-разрядного обмена данными между устройствами, памятью и процессором. Подсистема **PLUG & PLAY** полностью поддерживает PCI.

Posix (Portable Operation System Interface based on uniX, интерфейс для переносимых операционных систем, основанных на UNIX) — совокупность стандартов, включающая формальное описание программного интерфейса операционной системы, описание командной оболочки, политики безопасности и т. п. Приняты для решения проблем несовместимости различных коммерческих версий ОС UNIX — SunOS (в дальнейшем Solaris), HP-UX, Irix, AIX и пр.

PostScript (часто сокращенно называемый «PS») — стандарт де-факто печати в Unix-системах. Разработан компанией Adobe и лицензирован производителям принтеров и разработчикам программного обеспечения. Одной из наиболее известных в мире свободных ПО является Ghostscript, универсальный обработчик PS.

Pwd (команда Unix) — вывод имени текущего каталога (выведет регистрационный каталог, если мы в нем находимся).

Tty (команда Unix) — вывод информации о терминале, соединенном со стандартным вводом.

WIMPD (Windows, Menu, Pointng Device — окна, меню, указывающее устройство) — аббревиатура, обозначающая графические интерфейсы (как перечень основных «действующих лиц» в подобном интерфейсе).

Who (команда Unix) — вывод списка пользователей, подключенных в данный момент к системе, даты и времени входа каждого пользователя в систему.

Widget — графический элемент деталей рабочего стола, такой, как полоса прокрутки, кнопка или поле ввода, из которых складывается пользовательский интерфейс графической среды KDE, Windows и пр.

Автоматическое распределение — выделение памяти под данные в стеке. Такие данные существуют на протяжении работы текущей подпрограммы (функции или процедуры), затем уничтожаются.

Активизация системы — реализация процесса (программы).

Атрибуты файлов — каждый файл (каталог) имеет атрибут, который указывает на то, что этот файл является именно файлом, или на то, что он является каталогом. Атрибут файла — только для чтения Read only, скрытый Hidden, системный System, архивированный Archive.

Библиотеки объектных модулей — пакет объектных модулей, собранных в один файл и подключаемых к программе на этапе разрешения внешних ссылок (все идентификаторы, которые должны быть доступны из библиотек, объявляются в модулях как PUBLIC). Компоновщик может просматривать библиотеку и самостоятельно находить нужные модули, избавляя от этого программиста. Библиотеки делаются с помощью программы-библиотекаря.

Винчестер — несъемный жесткий магнитный диск (пакет дисков).

Виртуальная память (ВП) отличается от обычной ОП тем, что какие-то ее редко используемые фрагменты могут находиться на диске и подгружаться в реальную ОП по мере необходимости. Такая организация памяти позволяет снять ограничение, накладываемое объемом физической памяти, установленной на ЭВМ.

Внешние команды MS DOS — отдельные программы, которые для выполнения загружаются командным процессором MS DOS в ОП.

Внешние устройства — устройства ввода и вывода информации. Поскольку, как правило, они работают значительно медленнее остальных, управляющее устройство должно приостанавливать программу для завершения операции ввода-вывода с соответствующим устройством.

Внутренние команды MS DOS — команды, которые выполняются непосредственно командным процессором MS DOS.

Время оборота (turnaround time) — критерий эффективности планирования, измеряемый интервалом от момента появления процесса во входной очереди до момента его завершения. Это время названо временем оборота и включает время ожидания во входной очереди, время ожидания в очереди готовых процессов, время ожидания в очередях к оборудованию, время выполнения в процессоре и время ввода-вывода.

Время ожидания (waiting time) — критерий эффективности планирования, под которым понимается суммарное время нахождения процесса в очереди готовых процессов.

Время отклика (response time) — критерий эффективности планирования для сугубо интерактивных программ — время, прошедшее от момента попадания процесса во входную очередь до момента первого обращения к термину.

Готовность процесса. В распоряжении процесса (программы) имеются все ресурсы, кроме процессора.

Дескриптор (OC Unix) — уникальное целое положительное число, которое ставится в соответствие системой файлу при открытии. В процессе работы дескриптор используется процессом или его потомком для указания конкретного объекта операции.

Дескриптор процесса — динамика процесса определяется динамическими характеристиками дескриптора. В нем отображаются динамически изменяемые связи процесса с другими процессами. Все процессы, находящиеся в текущий момент в одном и том же состоянии, объединяют через дескрипторы в одну списковую структуру, но с учетом приоритетов процессов.

Динамическое выделение — выделение памяти под данные самой программой, когда это необходимо. Время жизни таких данных зависит от программы.

Дисциплина обслуживания FIFO (First In First Out) — обслуживание в порядке поступления: первый пришел — первый обслуживается. Все заявки на обслуживание поступают в конец очереди. Первыми обслуживаются заявки, находящиеся в начале очереди.

Дисциплина обслуживания LIFO — обслуживание заявок в порядке, обратном порядку поступления: последний пришел — первый обслуживается (Last In First Out). Является основой для построения стековой памяти.

Дисциплина распределения ресурса определяет порядок использования многими процессами того или иного ресурса, который в каждый момент времени может обслуживать только один процесс.

Долгосрочное планирование. На данном уровне объектом является не отдельный процесс, а некоторое объединение процессов по функциональному назначению, которое называется *работой* (приложением). Каждая работа рассматривается как независимая от других работ деятельность, связанная с использованием одной или многих программ и характеризующаяся конечностью и определенностью. По мере порождения новых работ создается собственная виртуальная машина для их выполнения. Планирование реализует программа ОС *долгосрочный* (планировщик). В OS/360 долговременный планировщик назывался *планировщиком заданий*.

Дорожка — концентрическая окружность на магнитной поверхности диска, где располагается информация. Дорожки нумеруются с 0-й (дорожка с самым большим радиусом).

Зависимые переключатели (радиокнопки, RadioButton) — группа переключателей для выбора одного из нескольких возможных взаимоисключающих режимов работы. Описания режимов находятся справа от кружков. В одной группе может быть включен только один из переключателей, остальные автоматически сбрасываются. Включенный (активный) режим индицируется точкой внутри кружка.

Задача — одна или несколько программ, связанных общим назначением, ресурсами.

Защита памяти осуществляется путем блокировки доступа к памяти других процессов, а также блокировки доступа к памяти ядра. Один из способов — вся память делится на страницы, и у каждой есть замок — 4-битовый признак, который можно установить только привиле-

гированной командой. В процессоре есть 4-битовый регистр — ключ, который также можно установить только привилегированной командой. При обращении происходит сравнение замка и ключа. С появлением многозадачности появилась проблема распределения памяти. При работе реальной программы обращения к ОП имеют тенденцию к локализации. Память можно разделить на используемую и неиспользуемую. Чтобы отследить использование области памяти, всю ОП можно разбить на страницы фиксированного размера (4К) и с каждой страницей связать бит, который устанавливают при обращении к данной странице.

Защита программ и данных в многозадачных ОС осуществляется с целью, чтобы сбой одной из выполняющихся программ не вызовел повреждения данных или кода других программ, и по возможности изолировать процессы друг от друга. Во всех ОС существуют хотя бы 2 режима процессора — *системный* и *пользовательский*. Переключение режимов работы осуществляется системными вызовами. Системный вызов — специальная команда, приводящая к прерыванию, и в ядре ОС существует несколько точек, куда перейдет управление по этому прерыванию.

Защищенный режим (protected mode) — режим работы процессора Intel 386, при котором он выполняет множество проверок корректности обращений к памяти, вызовов функций, доступа к портам ввода-вывода и т. д. Такая защищенность позволяет операционной системе обрабатывать ошибочные операции. Для того чтобы иметь возможность использовать все адресное пространство и преимущества виртуальной памяти процессора 386, приложение должно работать в *защищенном режиме*.

Идентификатор группы процессов (OC Unix). Каждый активный процесс является членом какой-либо группы процессов. В качестве идентификатора группы процессов используется идентификатор процесса, старшего в группе (общего родителя всех процессов группы). Объединение процессов в группу позволяет работать с группой, как с одним объектом, например передавать сигнал всем процессам группы. В некоторых версиях ОС Unix это свойство используется в языке управления заданиями.

Идентификатор группы терминала (OC Unix). Каждый активный процесс является членом группы, управляемой с конкретного терминала. Группа идентифицируется целым положительным числом, называемым «идентификатором группы терминала». Этот вид объединения процессов используется для управления доступом различных процессов к одному и тому же устройству.

Идентификатор процесса PID (OC Unix). Каждый активный процесс в системе идентифицируется уникальным целым положительным числом, называемым «идентификатором процесса». Диапазон представления идентификатора процесса — от 0 до ProcMax (значение ProcMax устанавливается при генерации системы).

Идентификатор процесса-предка PPID (OC Unix) — идентификатор процесса, породившего данный процесс (посредством вызова fork).

Имя файла (OS Unix) — последовательность от одного до максимально допустимого числа символов, используемая для именования обычных файлов, каталогов или специальных файлов. В имени файла допустимы любые символы кода ASCII, за исключением управляющего кода 0 (ПУС) и символа «/». Не рекомендуется использовать в именах файлов символы, имеющие специальное значение для языков управления заданиями (типа «*», «?»).

Исполняемый модуль — модуль, содержащий готовую к выполнению программу; может быть двух видов: *точный образ памяти* программы с привязкой к абсолютным адресам (в MS DOS — формат файла *.com) и *перемещаемый* исполняемый формат.

Исходный код программы — код, написанный на языке программирования. Может включать модули на ЯВУ и модули с подпрограммами на языке ассемблера.

Каталог (OS Unix) — специальный тип файла, содержащий информацию о файлах, которые могут адресоваться из данного каталога без указания полного имени (т. е. по имени файла). Любой каталог содержит по крайней мере два имени — «.» и «..». Они соответствуют данному каталогу («.») и каталогу, в который данный каталог входит («..»).

Кластер — минимальная порция информации, которую MS DOS считывает/записывает за одно обращение к диску. Кластер включает только последовательно расположенные секторы (цель — увеличить скорость обмена с диском). Размер кластера = N . Размер сектора = $N \cdot 512$ байт, где $N = 2, 4, 8$ и т. д.

Корневой и текущий (рабочий) каталоги (OS Unix). Каждому процессу поставлены в соответствие корневой и рабочий каталоги, использующиеся для поиска конкретных файлов по их именам. Корневой каталог процесса может не совпадать с корневым каталогом системы.

Краткосрочное планирование. На данном уровне объектом управления являются *процессы*, которые выступают как потребители центрального процессора для внутренних процессов или внешнего процессора для внешних процессов. Планирование осуществляет *краткосрочный* (short term scheduler / CPU scheduler). В OS/360 краткосрочный назывался *супервизором задач*.

Куча. Для реализации динамических структур данных используют так называемую кучу (heap). Это объем памяти, в котором можно выделить участок для произвольного элемента данных. Для кучи есть две операции: выделения памяти `ALLOCATE` и освобождения `FREE`. Эти функции не осуществляют никаких действий с собственно памятью. При выделении программист получает адрес, а при освобождении доступный объем кучи становится больше. Одного адреса для этих операций недостаточно, требуется еще и размер элемента данных. Если указатель **типизированный**, размер будет получен автоматически. В случае **нетипизированного** указателя размер должен быть передан в функцию. Для реализации кучи ЯВУ снабжаются диспетчерами памяти, которые выделяют и освобожда-

ют память, имеют сведения о ее фрагментации, знают наибольший фрагмент свободной памяти и ее общее количество и т. п. При ненадобности память должна своевременно освобождаться. При использовании динамической памяти возможна ситуация образования «мусора» — кусков памяти, на которые утеряны ссылки, но которые не были своевременно освобождены, поэтому менеджер считает их занятыми. Для оптимизации известна процедура «сбора мусора» — перестройки динамических структур с освобождением памяти из-под тех данных, на которые отсутствуют ссылки.

Кэш-память — сверхоперативная память, обращение к которой намного быстрее, чем к оперативной, и в которой хранятся наиболее часто используемые участки последней. При обращении к памяти сначала нужные данные ищутся в кэш-памяти. При их отсутствии производится обращение к оперативной памяти, в результате общее время доступа к памяти сокращается.

Микроядро — элемент клиент-серверной архитектуры ОС. Обычное ядро — это набор подпрограмм, предназначенных для организации работы ОС. В него входят функции управления памятью, управления процессами, работы с внешними устройствами. Подход микроядра заключается в том, чтобы в составе самого ядра оставить минимальный набор подпрограмм, а все остальные функции возложить на отдельные модули или сервисы.

Модульное программирование — технология программирования на основе разбиения программы на подпрограммы по специфике обрабатываемых данных. Для этой цели в ЯВУ используются функции и процедуры. При вызове подпрограммы в стеке сохраняется текущее значение счетчика команд (ближняя модель вызова) и значение сегмента кода (дальняя модель вызова). При использовании дальней модели вызова подпрограмма необязательно должна находиться в том же сегменте, что и вызывающая программа. По окончании выполнения кода подпрограммы эти данные восстанавливаются, управление передается на следующий оператор после оператора вызова процедуры. До вызова подпрограммы в стек помещаются параметры — аргументы подпрограммы. Если передан не сам аргумент, а его адрес, то подпрограмма может изменить аргумент, в противном случае нет, так как по завершении работы подпрограммы стек очищается от аргументов. Отличие функций от процедур в том, что функции могут возвращать значения в вызывающую программу и их можно присваивать, например какой-либо переменной. В действительности функция при возврате значений просто модифицирует регистры процессора, а ЯВУ по соглашению использует данные из этих регистров.

Наиболее подходящего стратегия (best fit strategy) — выбор процесса из очереди при освобождении раздела памяти. Выбирается процесс, которому в освободившемся разделе наиболее тесно (выигрыш в памяти).

Наименее подходящего стратегия (last fit strategy) — выбор процесса из очереди при освобождении раздела памяти. Выбирается процесс, которому в освободившемся разделе наиболее свободно (в этом случае ос-

тающийся фрагмент часто достаточен для размещения еще одного процесса).

Объект (object) — формально это совокупность данных и методов работы с ними, некоторые из которых могут использоваться другим приложением. Объектно-ориентированные технологии позволяют создателю объекта определить интерфейсы к возможностям объекта, скрыв при этом особенности его реализации. Это делает возможным использование объекта многими непосредственно не относящимися к нему приложениями. Несмотря на то, что этот термин широко используется в Windows 95, в большинстве случаев он применяется в значении «данные» или «нечто». Слово «объект» — это, пожалуй, самый заметный кандидат на звание наиболее перегруженного термина в области программного обеспечения.

Объектный модуль — код программы после трансляции (компиляции), преобразованный в машинные коды. Помимо них содержит внешние ссылки и информацию для редактора связей и может также содержать отладочную информацию (debug info).

Ожидание процесса — ожидание всех ресурсов для развития процесса (программы).

Окончание процесса — высвобождение всех ресурсов процессом.

Оперативная память предназначена для хранения программ и данных, которыми они манипулируют. Физически выполнена в виде некоторого числа микросхем. Логически ОП можно представить как линейную совокупность ячеек, каждая из которых имеет свой номер, называемый адресом. Время записи и чтения из ОП в современных машинах занимает доли микросекунды, а для других устройств это время в 10—1000 раз больше. Число микросхем памяти, физически присутствующих в компьютере, определяет объем памяти, которую можно использовать для программ и данных. Это число может меняться от машины к машине. Объем памяти обычно можно наращивать с помощью плат расширения, вставляемых в специальные разъемы. Для процессора память — это не более чем несколько тысяч 8-разрядных ячеек, каждая из которых имеет уникальный адрес.

Операционная оболочка — программа, один из модулей которой (резидентный) постоянно находится в памяти компьютера и для выполнения каких-либо заданных пользователем функций загружает с диска в свободные области памяти необходимые исполнительные модули. Операционные оболочки предназначены в основном для упрощения выполнения команд ОС и удобного представления требуемой информации.

Операционная система обеспечивает следующие функции — управление процессором путем передачи управления программам, обработку прерываний, синхронизацию доступа к ресурсам, управление памятью, управление устройствами ввода-вывода, управление инициализацией программ, межпрограммные связи, управление данными на долговременных носителях путем поддержки файловой системы.

Организация ввода-вывода в современных ЭВМ осуществлена с использованием *прерываний*. Это связано с тем, что устройства ввода-вывода работают намного медленнее, чем процессор и оперативная память. Поэтому управляющее устройство должно приостанавливать выполнение программы и ждать завершения операции ввода-вывода с внешним устройством. При выводе все результаты выполненной программы должны быть выведены на ВУ, после чего процессор переходит к ожиданию сигналов от ВУ. При вводе, например, с клавиатуры получение значений нажатых клавиш осуществляется при поступлении прерывания от клавиатуры.

Очередь готовых процессов (*ready queue*) — готовые к выполнению процессы, расположенные в основной памяти и ожидающие освобождения ресурса «процессорное время».

Очередь к оборудованию ввода-вывода (*devices queue*) — одна из очередей, в которой находится процесс при ожидании завершения операций ввода или вывода.

Очередь работ/заданий (*job queue*) — входная очередь для новых процессов.

Пейджинг — механизм виртуальной памяти, при котором страницы вытесняются на диск и подкачиваются с диска.

Первого подходящего стратегия (*first fit strategy*) — выбор процесса из очереди при освобождении раздела памяти. Выбирается первый процесс, который может разместиться в освободившемся разделе.

Планировщик — программа, выполняющая алгоритм планирования процессов. Планирование очередности предоставления выполняющимся процессам времени центрального процессора (диспетчеризация). Процессы работают с центральным процессором в режиме разделения времени.

Поле ввода — область, в которую пользователь может вводить информацию с клавиатуры. В этой области указатель мыши принимает новую форму. Если в этот момент щелкнуть кнопкой мыши, то в поле появится курсор и можно вводить данные.

Поле ввода с раскрывающимся списком — это комбинация элементов поля ввода и раскрывающегося списка. Такой элемент позволяет как непосредственно (вручную) вводить данные в поле ввода, так и заполнить его значением из раскрывающегося списка.

Поле ввода со счетчиком обычно используется для ввода числовых значений. Его можно заполнить как обычное поле ввода или воспользоваться кнопками, расположенными справа. В этом случае значение в поле будет изменяться (соответственно увеличиваться и уменьшаться) с наиболее оптимальным шагом и при этом не превысит предельных значений.

Полное имя файла (OC Unix) — последовательность имен каталогов, разделенных символами «/», предшествующая имени файла. Полное имя файла содержит информацию о положении каталога с файлом в дереве

файлов. Если полное имя начинается с символа «/» (абсолютная адресация), оно указывает положение каталога с файлом относительно корня дерева файлов. Имя, начинающееся с любого другого символа, указывает положение каталога с файлом относительно текущего каталога. Длина полного имени файла не может превышать (PathnameMax) символов.

Порождение процесса — создание условий для реализации программы.

Поток (мини-задача, шаг, цепь, нить) — это последовательности команд процесса, которые выполняются независимо одна от другой и используют общие ресурсы одного процесса.

Прерывания — специфические сигналы, посылаемые процессору устройством или программой, когда требуется его немедленное вмешательство. В этом случае он останавливает всякую другую деятельность и вызывает программу *обработчик прерывания*. По окончании ее работы он продолжает прерванную работу с того места, где она остановилась. Прерывания бывают двух типов — *аппаратные* (генерируются схемами ПК в ответ на какое-либо действие, например при нажатии клавиши на клавиатуре генерируется прерывание), иногда аппаратные прерывания генерируются устройством в случае некорректной работы программы, например деление на 0; *программные* — генерируются программой для вызова различных подпрограмм из ОП и ПЗУ.

Привилегированный пользователь и привилегированный процесс (OS Unix). Процесс считается привилегированным, т. е. получает исключительные права доступа ко всем ресурсам, если его эффективный идентификатор пользователя равен нулю.

Пропускная способность. Пропускная способность процессора измеряется количеством заданий, которые выполняются в единицу времени.

Процесс — минимальный программный объект, обладающий собственными системными ресурсами (запущенная программа). Процесс, как любая деятельность по исполнению программы на процессоре, нуждается в управлении, которое заключается в переводе его из одного состояния в другое: порождение — готовность — активизация системы — ожидание — окончание.

Процесс выполняемый (running) — команды программы выполняются процессором.

Процесс готовый (ready) — процесс ожидает освобождения процессора ЭВМ.

Процесс завершённый (terminated) — процесс завершил свою работу.

Процесс новый (new) — процесс только что создан.

Процесс ожидающий (waiting) — процесс ожидает завершения некоторого события, чаще всего операции ввода-вывода.

Процессор (ЦП, CPU) — устройство, выполняющее вычислительные операции и управляющее работой ЭВМ. Содержит *устройство управления*, выбирающее машинные команды из памяти и выполняющее их, и

арифметико-логическое устройство, выполняющее арифметические и логические операции. Работа всех электронных устройств машины координируется сигналами, вырабатываемыми ЦП. В современных ПК процессор представлен одной СБИС, содержащей свыше миллиона транзисторов.

Работа — объединение процессов (задач), рассматриваемых как единое целое в операционной системе (проект).

Раскрывающийся список (List). При нажатии на пиктограмму со стрелкой открывается список всех возможных значений, которые можно выбрать для установок в этом элементе. Если список длинный, то появится линейка прокрутки, с помощью которой можно просмотреть все элементы списка.

Реальные идентификаторы пользователя и группы UID, GID (ОС Unix). Каждый пользователь системы идентифицируется целым положительным числом, называемым «идентификатором пользователя». В то же время пользователь может являться членом одной или нескольких групп. Группы отличаются друг от друга специфическими правами доступа. Положительное целое число, поставленное в соответствие группе, называется «идентификатором группы». Процесс имеет реальные идентификаторы пользователя и группы, значения которых равны соответствующим идентификаторам пользователя, инициировавшего данный процесс. Реальные идентификаторы наследуются всеми потомками процесса.

Реальный режим (real mode) — режим работы процессора Intel 386, совместимый с процессором Intel 8086. В реальном режиме невозможны доступ к огромному *виртуальному адресному пространству 386* процессора или такие возможности, как, например, *замещение страниц по требованию*.

Регулятор используется для установки параметров от минимального до максимального с помощью движка.

Редактирование связей — разрешение внешних ссылок и создание исполняемого модуля из совокупности объектных. Основные функции редактора связей — распределение памяти, разрешение внешних ссылок.

Режим доступа файла (ОС Unix) определяет права процесса на выполнение операций над файлом (например, открытие файла на запись), является принадлежностью файла и устанавливается при его создании, может переопределяться с помощью вызова **chmod**. Режим доступа содержит информацию о возможности чтения, записи и выполнения файла для трех групп пользователей: владельца файла, членов его группы и всех остальных пользователей. Для каталогов признак «выполнения» интерпретируется как право просмотра его содержимого.

Резидентная программа — программа, которая после загрузки в ОП и передаче ей управления инициализируется таким образом, что постоянно находится в ОП и выполняется параллельно другим программам.

Ресурс (resource) — сетевой *объект*, такой, как принтер или набор связанных в каталог файлов, доступный для совместного использования.

Свопинг — алгоритм реализации виртуальной памяти. Его можно разбить на три части: управление пространством на устройстве выгрузки, выгрузка процессов из основной памяти и подкачка процессов в основную память. В качестве устройства выгрузки используют раздел на устройстве типа жесткого диска или дисковый файл (swap-file) на таком устройстве.

Сегментная виртуальная память использует сегментно-страничное виртуальное адресное пространство. При этом требуется явное выделение частей программы. Реальная память также подвергается сегментной детализации по модульному признаку.

Сегментно-страничная виртуальная память. Каждый модуль программы подвергается дополнительному страничному структурированию. При этом размеры получаемых сегментов могут быть увеличены. В этом случае оперативная память подвергается также страничной структуризации. Такая схема реализует компромисс между операционной системой и пользователем, а именно: операционная система увеличивает пропускную способность, используя преимущества страничной по запросу схемы функционирования оперативной памяти, а пользователь избавлен от ограничений на размеры разрабатываемой программы.

Сектор. Каждая дорожка, размещенная на диске, делится на секторы (блоки). Каждый сектор имеет размер 512 байт (для MS DOS).

Сервер (server) — сетевой компьютер, на котором находятся доступные клиентам ресурсы. Ресурсами сервера могут быть файлы, принтеры или приложения сервера (такие, как многопользовательские базы данных).

Сети типа «клиент-сервер» (client-server networking) — сетевая архитектура, в которой предназначенные для совместного использования ресурсы (resources) сосредоточены на мощных компьютерах — серверах (server machines), а подключенные к ним настольные машины играют роль клиентов (clients), посылая по сети запросы на ту или иную информацию.

Системное программное обеспечение обеспечивает интерфейс между программистом или пользователем и аппаратной частью ЭВМ (операционная система, программы-оболочки), выполняет вспомогательные функции (программы-утилиты).

Системные вызовы (OS Unix) представляют собой интерфейс между программами пользователя и ядром операционной системы Unix. Реализация системных вызовов — одна из функций ядра OS Unix.

Системные процессы (OS Unix). Процессы с идентификаторами 0, 1 считаются системными. Это: планировщик (процесс 0), инициализирующий процесс, который одновременно является родителем всех остальных процессов (процесс 1).

Смежное размещение (contiguous allocation) — размещение программ в памяти, предполагающее, что в памяти, начиная с некоторого начального адреса, выделяется один непрерывный участок адресного про-

странства. При *несмежном размещении* программа разбивается на множество частей, которые располагаются в различных, не обязательно смежных участках адресного пространства.

Список — элемент, содержащий все возможные в каждом конкретном случае значения, которые пользователь может установить, добавить или изменить эти значения непосредственно в списке нельзя.

Статическое выделение памяти — выделение памяти под информацию внутри сегмента данных программы. Такие данные существуют на протяжении всей жизни программы до ее завершения.

Стек — среда для размещения данных для возврата из подпрограмм, а также их аргументы и автоматические данные. Все это может потребовать достаточно большого размера стека. Как правило, программист может определять размер стека в программе.

Страничная организация памяти — организация, при которой адресное пространство памяти разбивается на малые участки — страницы. Используется для управления памятью в системах, работающих в защищенном режиме. Как правило, такая организация памяти подразумевает *пейджинг*.

Страничная по запросу виртуальная память снимает требование полного расположения программы в оперативной памяти, т. е. в оперативной памяти может размещаться блок программы, и снимает ограничение на размер виртуальной памяти. Отображение происходит динамически и по частям.

Супервизор — программа многозадачной ОС, обеспечивающая наилучшее использование ресурсов ЭВМ при одновременном выполнении нескольких задач.

Таблица управления процессом (PCB — process control block). В PCB процесс описывается набором значений, параметров, характеризующих его текущее состояние и используемых операционной системой для управления прохождением процесса через компьютер.

Трансляция — получение объектного кода из исходного.

Управление процессами обеспечивает повышение производительности операционной системы за счет организации параллельной работы процессора с внешними устройствами различного быстродействия. Решение этой задачи связано с управлением памятью, так как процесс может развиваться только в оперативной памяти. Реализация управления процессами требует дополнительных ресурсов времени и памяти. Управление процессами в любой операционной системе реализуется с помощью специальной структуры — дескриптора процесса, которая содержит основную информацию о процессе.

Управляющие кнопки (Button) предназначены для выполнения действий. Какое именно действие выполняет кнопка, написано непосредственно на ней. Кнопка приводится в действие нажатием мыши на ней. Если в конце названия кнопки присутствуют три точки, то такая кнопка вызовет новое диалоговое окно.

Утилизация (использование) CPU (utilization) — критерий эффективности планирования. Утилизация CPU теоретически может находиться в пределах от 0 до 100 %. В реальных системах утилизация CPU колеблется в пределах 40—90 %.

Флажок (или Независимый переключатель, CheckBox) — переключатель для режима работы, описание которого находится справа от квадрата. Он может быть включен (установлен) — внутри квадрата изображен значок, или выключен (сброшен) — внутри пусто. Для установки или сброса флажка необходимо щелкнуть мышью в квадрате или на его описании. Такой элемент вполне самостоятельно определяет свой параметр и поэтому называется независимым, в отличие от следующего элемента.

Флоппи-диск (дискета) — съемный гибкий магнитный диск.

Цилиндр — объединение дорожек с одним и тем же номером, расположенных на разных поверхностях диска (для флоппи-диска под цилиндром подразумеваются две дорожки). Цилиндр — пространство, доступное для записи-считывания при фиксированном положении блока головок дисководов.

Чисто страничная виртуальная память — схема виртуальной памяти, основанная на том, что выделяется вся необходимая память для реализации программы или блока программ, т. е. ставится условие, что программа целиком располагается в оперативной памяти.

Шина (bus) — устройство для организации интерфейса с другими устройствами. К шине подключаются платы адаптеров.

Эффективные идентификаторы пользователя и группы, список групп доступа (OS Unix). Доступ к системным ресурсам определяется тремя значениями: «эффективным идентификатором пользователя», «эффективным идентификатором группы» и «списком групп доступа». Эффективные идентификаторы пользователя и группы при старте процесса обычно совпадают с реальными. Исключение составляет случай, когда статус выполняемого файла содержит **признаки set-UID** и **set-GID** (см. описание вызова **execve**). Список групп доступа вместе с идентификатором группы используется для определения прав доступа к системным ресурсам.

Ядро ОС Unix — программа, которая обеспечивает разделение времени центрального процессора между выполняющимися процессами; осуществляет управление памятью и устройствами ввода-вывода, реализует файловую систему.

Команды и функции FAR MANAGER

Таблица П1.1. Редактирование командной строки

Действие	Команда (клавиши)
Символ влево	<←>, <Ctrl+S>
Символ вправо	<→>, <Ctrl+D>
Слово влево	<Ctrl+←>
Слово вправо	<Ctrl+→>
В начало строки	<Ctrl+Home>
В конец строки	<Ctrl+End>
Удалить символ	<Del>
Удалить символ слева	<BS>
Удалить до конца строки	<Ctrl+K>
Удалить слово слева	<Ctrl+BS>
Удалить слово справа	<Ctrl+Del>
Рассматривать следующую комбинацию клавиш как код	<Ctrl+Q>
Копировать в Буфер Обмена	<Ctrl+Ins>
Вставить из Буфера Обмена	<Shift+Ins>
Предыдущая команда	<Ctrl+E>
Следующая команда	<Ctrl+X>
Очистить командную строку	<Ctrl+Y>
Вставить имя файла из активной панели	<Ctrl+Enter>
Вставить полное имя файла из активной панели	<Ctrl+F>
Вставить путь из левой панели	<Ctrl+[>
Вставить путь из правой панели	<Ctrl+]>
Вставить путь из активной панели	<Ctrl+Shift+[>
Вставить путь из пассивной панели	<Ctrl+Shift+]>

Примечания:

1. Если командная строка пуста, <Ctrl+Ins> будет копировать имена выбранных в панели файлов в Буфер Обмена так же, как и <Ctrl+Shift+Ins>.

2. <Ctrl+End>, нажатая в конце командной строки, заменяет ее текущее содержимое командой из Истории Команд, начинающейся с уже введенных букв, если такая команда существует. Чтобы перейти к следующей такой команде, вы можете нажать <Ctrl+End> повторно.

3. Большинство из описанных выше команд действительно для всех строк редактирования, включая строки в диалогах и встроенный редактор.

Таблица П1.2. Общие команды управления панелями

Действие	Команда (клавиши)
Изменить активную панель	<Tab>
Поменять панели местами	<Ctrl+U>
Перечитать содержимое панели	<Ctrl+R>
Убрать/показать информационную панель	<Ctrl+L>
Убрать/показать панель быстрого просмотра	<Ctrl+Q>
Убрать/показать дерево папок	<Ctrl+T>
Убрать/показать обе панели	<Ctrl+O>
Убрать/показать неактивную панель	<Ctrl+P>
Убрать/показать левую панель	<Ctrl+F1>
Убрать/показать правую панель	<Ctrl+F2>
Изменить высоту панелей	<Ctrl+↑>, <Ctrl+↓>
Изменить ширину (при пустой командной строке)	<Ctrl+←>, <Ctrl+→>
Восстановить ширину панелей по умолчанию	<Ctrl+Num5>

Таблица П1.3. Команды файловой панели

Действие	Команда (клавиши)
Пометить/снять пометку файла	<Ins>, <Shift+Клавиши курсора>
Пометить группу	<Num+>
Снять пометку с группы	<Num->
Инвертировать пометку	<Num*>
Пометить файлы с расширением как у текущего файла	<Ctrl+Num+>
Снять пометку с файлов с расширением как у текущего	<Ctrl+Num->
Инвертировать пометку, включая папки	<Ctrl+Num*>

Окончание табл. П1.3

Действие	Команда (клавиши)
Пометить файлы с именем как у текущего файла	<Alt+Num+>
Снять пометку с файлов с именем как у текущего файла	<Alt+Num->
Пометить все файлы	<Shift+Num+>
Снять пометку со всех файлов	<Shift+Num->
Поместить помеченные имена в Буфер Обмена	<Ctrl+Shift+Ins>
Восстановить предыдущую пометку	<Ctrl+M>
Прокрутка длинных имен и описаний	<Alt+←>, <Alt+→>
Установить краткий режим просмотра	<ЛевыйCtrl+1>
Установить средний режим просмотра	<ЛевыйCtrl+2>
Установить полный режим просмотра	<ЛевыйCtrl+3>
Установить широкий режим просмотра	<ЛевыйCtrl+4>
Установить детальный режим просмотра	<ЛевыйCtrl+5>
Установить режим просмотра описаний	<ЛевыйCtrl+6>
Установить режим просмотра длинных описаний	<ЛевыйCtrl+7>
Установить режим просмотра владельцев файлов	<ЛевыйCtrl+8>
Установить режим просмотра связей файлов	<ЛевыйCtrl+9>
Установить альтернативный полный режим просмотра	<ЛевыйCtrl+0>
Убрать/показать файлы с атрибутом Скрытый и Системный	<Ctrl+H>
Переключить вывод длинных/коротких имен файлов	<Ctrl+N>
Спрятать/показать левую панель	<Ctrl+F1>
Спрятать/показать правую панель	<Ctrl+F2>
Сортировать файлы активной панели по имени	<Ctrl+F3>
Сортировать файлы активной панели по расширению	<Ctrl+F4>
Сортировать файлы активной панели по времени модификации	<Ctrl+F5>
Сортировать файлы активной панели по размеру	<Ctrl+F6>
Не сортировать файлы активной панели	<Ctrl+F7>
Сортировать файлы активной панели по времени создания	<Ctrl+F8>
Сортировать файлы активной панели по времени доступа	<Ctrl+F9>
Сортировать файлы активной панели по описаниям	<Ctrl+F10>
Сортировать файлы активной панели по владельцу	<Ctrl+F11>
Вывести меню режимов сортировки	<Ctrl+F12>
Использовать сортировку по группам	<Shift+F11>
Показывать помеченные файлы первыми	<Shift+F12>

Примечания:

1. Если командная строка пуста, <Ctrl+Ins> будет копировать меню выбранных в панели файлов в Буфер Обмена так же, как и <Ctrl+Shift+Ins>.

2. Если включена опция Разрешить обратную сортировку в диалоге Настроек панели, то повторное нажатие одной и той же клавиши сортировки файлов приводит к смене направления сортировки с возрастающей на убывающую и наоборот.

3. Комбинации <Alt+→> и <Alt+←>, используемые для прокрутки длинных имен и описаний, работают только с клавишами <→> и <←>, расположенными на основной (левой) части клавиатуры. Это связано с тем, что при нажатой <Alt> клавиши управления курсором на Num используются для ввода символов через их десятичные коды.

Таблица П1.4. Использование функциональной клавиатуры для управления файлами и выполнения сервисных команд

Действие	Клавиши	Примечания
Помощь	<F1>	
Вызвать Пользовательское меню	<F2>	
Просмотр	<Ctrl+Shift+F3>, <Num5>, <F3>	Если <Num5> или <F3> нажаты при курсоре
Редактирование, распаковка	<Ctrl+Shift+F4>, <F4>	<F4> вызывает встроенный редактор, внешний или ассоциированный с файлом в зависимости от типа файла и настроек редактора. <Ctrl+Shift+F4> всегда вызывает встроенный редактор вне зависимости от файловых ассоциаций
Копирование	<F5>	Копирует файлы и папки. Если требуется создать папку назначения перед копированием, следует добавить к ее имени обратную черту (\)
Переименование или перенос	<F6>	Переименование или перенос файлов и папок. Если вы хотите создать папку назначения перед переносом, добавьте к ее имени обратную черту (\)
Создание новой папки	<F7>	
Удаление	<Shift+Del>, <Shift+F8>, <F8>	Удаление файлов и папок. <F8> и <Shift+Del> удаляют все выбранные файлы, <Shift+F8> – только файл под курсором. <Shift+Del> всегда удаляет файлы, не используя Корзину (Recycle Bin). Использование Корзины командами <F8> и <Shift+F8> зависит от конфигурации
Уничтожение файлов и папок	<Alt+Del>	Уничтожает файлы и папки. Перед удалением файл перезаписывается нулями, усекается до нулевой длины и переименовывается во временное имя
Показать Горизонтальное Меню	<F9>	
Завершить работу с FAR	<F10>	
Показать команды Подключаемых Модулей (Plugins)	<F11>	

Таблица П1.5. Команды с использованием ФК и клавиш <Alt>, <Shift>

Действие	Клавиши	Примечания
Изменить текущий диск в левой панели	<Alt+F1>	
Изменить текущий диск в правой панели	<Alt+F2>	
Встроенная/внешняя программа просмотра	<Alt+F3>	Вызывает внешнюю программу просмотра, если по умолчанию используется внутренняя, и внутреннюю, если по умолчанию используется внешняя
Встроенный/внешний редактор	<Alt+F4>	Вызывает внешний редактор, если по умолчанию используется внутренний, и внутренний, если по умолчанию используется внешний
Печать файлов	<Alt+F5>	
Создать жесткую связь (только на NTFS)	<Alt+F6>	Используя жесткие связи файлов, можно иметь несколько различных имен файлов, ссылающихся на одни и те же данные
Поиск файла	<Alt+F7>	
Показать Историю Команд	<Alt+F8>	
Переключение между 25-й и 50-й строками на экране	<Alt+F9>	
Поиск папки	<Alt+F10>	
Показать историю просмотра и редактирования	<Alt+F11>	
Показать историю папок	<Alt+F12>	
Добавить файлы к архиву	<Shift+F1>	
Извлечь файлы из архива	<Shift+F2>	
Выполнить команды управления архивом	<Shift+F3>	
Редактировать новый файл	<Shift+F4>	
Копирование файла под курсором	<Shift+F5>	
Переименование или перенос файла под курсором	<Shift+F6>	
Удаление файла под курсором	<Shift+F8>	
Сохранить конфигурацию	<Shift+F9>	
Выбрать последний выполненный пункт меню	<Shift+F10>	

Таблица П1.6. Прочие команды

Действие	Команда (клавиши)
Запуск, смена папки, вход в архив	<Enter>
Запуск в отдельном окне	<Shift+Enter>
Сменить папку на корневую	<Ctrl+\>
Смена папки, вход в архив (также в SFX-архив)	<Ctrl+PgDn>
Перейти в папку уровнем выше	<Ctrl+PgUp>
Создать ссылку на текущую папку	<Ctrl+Shift+N>
Использовать ссылку на папку	<ПравыйCtrl+N>
Установить атрибуты файлов	<Ctrl+A>
Применить Команду к помеченным файлам	<Ctrl+G>
Добавить Описания к помеченным файлам	<Ctrl+Z>
Записать клавиатурную макрокоманду	<Ctrl+<.>
Очистка истории в строках редактирования диалогов	<Del>

Главное меню Konqueror

	Пункт	Клавиши	Действия
Меню Адрес			
Адрес	Новое окно	<Ctrl+N>	Открыть новое окно
	Новая вкладка	<Ctrl+Shift+N>	Открыть новую вкладку
	Создать копию окна	<Ctrl+D>	Открыть текущую страницу в новом окне
	Открыть страницу...	<Ctrl+O>	Открыть каталог или файл (следует указать путь к файлу или каталогу, например /home/tata или /home/tata/alex.txt)
	Отправить ссылку		Отправить ссылку на данную страницу по электронной почте
	Отправить файл		Отправить данный файл по электронной почте в виде приложения к письму
	Сохранить фон как...		Если на текущей странице есть фоновый рисунок, этот пункт позволяет сохранить его на компьютере
	Сохранить как...	<Ctrl+S>	При просмотре документа или Web-страницы эта команда позволяет сохранить ее
	Сохранить врезку как...		Данный пункт имеет ту же функцию, что и пункт Сохранить как..., но предназначен для работы с фреймами
	Печать...	<Ctrl+P>	Напечатать текущую страницу
	Печать врезки		Напечатать текущую врезку страницы
	Открыть в Netscape или Mozilla		Открыть текущую страницу в обозревателе Netscape или Mozilla
	Выход	<Ctrl+Q>	Закрыть текущее окно Konqueror

Продолжение прил. 2

	Пункт	Клавиши	Действия
Меню Правка (см. рис. 4.18) Большинство команд меню Правка также доступны в меню, появляющемся при щелчке правой кнопкой мыши на свободном пространстве окна			
Правка	Отменить	<Ctrl+Z>	Отменить последнее выполненное действие
	Вырезать	<Ctrl+X>	Поместить выделенные объекты в буфер обмена. После этого командой Вставить можно переместить объекты в новый каталог
	Копировать	<Ctrl+C>	Копировать выделенные объекты в буфер обмена
	Вставка	<Ctrl+V>	Вставить объекты из буфера в текущую открытую папку
	Переименовать	<F2>	Переименовать файл или каталог, не открывая окно Свойства...
	Выбросить в корзину	<Del>	Переместить выделенные объекты в корзину
	Удалить	<Shift+Del>	Удалить выделенные объекты
	Копировать файлы	<F7>	Копировать выделенные файлы в другой каталог
	Переместить файлы...		Переместить выделенные элементы в другой каталог
	Создать		Создать каталог, текстовый файл или файл HTML, ссылку на приложение, адрес URL, гибкий диск или компакт-диск. Подробнее об этом см. табл. 4.11
	Изменить тип файла...		Открыть окно, позволяющее изменить тип файла
	Свойства...		Открыть окно Свойства

Продолжение прил. 2

	Пункт	Клавиши	Действия
Правка	Правка\Выбранные\... Опции для работы с выбранными элементами в окне Konqueror		
	Выделить..	<Ctrl++>	Данная команда позволяет работать с множеством файлов путем их выделения. При выборе данного пункта появится окно, в котором следует ввести шаблон (используя специальные символы * и ?)
	Отменить выделение...	<Ctrl+->	Снять выделение с выбранных элементов
	Отменить все выделение	<Ctrl+U>	Снять выделение со всех файлов и каталогов
	Инвертировать выделение	<Ctrl+*>	Обратить выделение (все выделенные объекты становятся невыделенными, и наоборот)
	Выделить все	<Ctrl+U>	Выделить весь текст на странице (после этого вы можете воспользоваться командами Копировать и Вставить, чтобы скопировать текст в текстовый редактор)
	Найти...	<Ctrl+F>	Найти заданную строку на текущей текстовой или HTML-странице
	Найти далее	<F3>	Продолжить поиск заданной строки
Вид	Перейти к строке...		Перейти к определенной строке текущей страницы
	Меню Вид		
	Режим отображения...		Выбрать один из режимов отображения каталогов — Значки, Список, В виде дерева, Таблица или Только текст
	Использовать index.html		Если в каталоге имеется файл index.html, то при просмотре такого каталога вместо списка файлов будет показано содержимое этого файла
	Заблокировать в текущем месте		Заблокировать в текущем месте

Продолжение прил. 2

	Пункт	Клавиши	Действия
Вид	Разблокировать все панели		Разблокировать все панели
	Связать панель		Связать текущую панель с другими
	Обновить	<F5>	Обновить текущую страницу
	Остановить	<Esc>	Остановить загрузку страницы
	Размер пиктограмм		Выбрать размер значков, отображающихся в режиме Значки
	Сортировать по		Выбрать порядок, в котором значки отображаются в режимах Значки и Таблица
	Сортировка без учета регистра		Указать, будет ли учитываться регистр символов при сортировке в режимах В виде дерева, Список и Только текст
	Просмотреть исходный текст документа	<Ctrl+U>	Показать исходный текст страницы (только при просмотре документа или страницы HTML)
	Исходный текст врезки		Показать исходный текст текущей врезки (только при просмотре страницы, на которой используются врезки)
	Просмотреть сведения о документе	<Ctrl+X>	Просмотреть информацию о документе, такую, как заголовок, URL, заголовки HTTP (только при просмотре документа или страницы HTML)
	Безопасность...		Позволяет узнать, используется ли в данный момент безопасное соединение (с помощью SSL), а также открыть окно Настройка шифрования... Те же возможности доступны при щелчке левой кнопкой мыши по значку с изображением замка в меню
	Кодировка		Позволяет указать кодировку страницы. Чаще всего для правильного отображения страницы достаточно выбрать пункт Автоматически
	Предварительный просмотр		В качестве значков к файлам HTML, рисункам и текстовым файлам будет использоваться содержимое самих файлов в уменьшенном виде

Продолжение прил. 2

	Пункт	Клавиши	Действия
Вид	Показывать скрытые файлы		Следует ли отображать скрытые файлы (имена которых начинаются с точки)
	Показывать под-рбно...		Какие именно сведения о файлах и каталогах будут показываться в режимах Таблица, Только текст, В виде дерева
	Цвет фона...		Выбрать цвет фона основного окна (для менеджера файлов)
	Фоновый рисунок...		Выбрать фоновый рисунок (для менеджера файлов)
Меню Перейти			
Перейти	Вверх	<Alt+↑>	Подняться на один уровень в структуре каталогов
	Назад	<Alt+←>	Перейти к предыдущей странице
	Вперед	<Alt+→>	Перейти к следующей странице
	Домашняя страница	<Ctrl+Home>	Перейти в домашний каталог
	Приложения		Открыть папку, в которой находятся ссылки на установленные вами приложения
	Корзина		Открыть папку Корзина в отдельном окне
	Шаблоны		Открыть папку Шаблоны в отдельном окне
	Автозагрузка		Открыть папку Автозагрузка в отдельном окне
	Часто посещаемые...		Показывает наиболее часто посещаемые URL. При выборе одного из них он будет открыт в окне Konqueror
Меню Закладки			
Закладки	Добавить закладку	<Ctrl+B>	Добавить к закладкам ссылку на выделенный объект
	Сохранить вкладку в папке закладок...		Создает папку с закладками URL всех открытых вкладок в Konqueror

Продолжение прил. 2

	Пункт	Клавиши	Действия
Закладки	Управление закладками...		Открыть Редактор закладок
	Создать папку...		Создать в папке с закладками новую папку
Меню Сервис Если в системе установлены дополнительные модули Konqueror, в меню Сервис появятся еще несколько пунктов (см. табл. 4.16)			
Сервис	Выполнить команду...	<Alt+F2>	Выбрав этот пункт, вы сможете ввести команду, которую следует выполнить
	Открыть терминал	<Ctrl+T>	Открыть окно терминала (Konsole)
	Поиск файла...		Открыть KFind (программу поиска файлов)
Меню Настройка			
Настройка	Показывать панель меню	<Ctrl+M>	Показать или скрыть панель меню
	Панели		При выборе данного пункта появится меню, в котором вы сможете указать, какие панели следует показывать, а какие нет
	Полноэкранный режим	<Ctrl+Shift+F>	Переводит Konqueror в полноэкранный режим без обычных элементов окна. Для того чтобы выйти из полноэкранного режима, нажать кнопку Выход из полноэкранного режима в панели инструментов или <Ctrl+Shift+F>
	Сохранить параметры просмотра папки		Сохраняет параметры представления текущей папки. При этом в папке будет создан файл .directory, в котором будут сохранены параметры просмотра папки. При повторном открытии папки Konqueror эти параметры будут восстановлены
	Удалить свойства папки		Удалить параметры просмотра папки, сохраненные при выборе Сохранять настройки просмотра для папки
	Загрузить профиль просмотра		Загрузить параметры профиля просмотра

Продолжение прил. 2

	Пункт	Клавиши	Действия
Настройка	Сохранить профиль просмотра "Имя-Профиля"...		Сохранить параметры текущего профиля просмотра
	Настроить профили просмотра...		Данный пункт позволяет изменить существующий профиль или создать новый
	Настройка клавиатуры...		Настроить привязки клавиш, т. е. какие действия будут выполняться при нажатии того или иного сочетания клавиш (например, нажатие <Ctrl+V> будет соответствовать действию Копировать)
	Настройка панелей...		Данный пункт позволяет изменить внешний вид основной и дополнительной панелей, а также панели адреса
	Настроить Konqueror...		Данный пункт позволяет изменить различные параметры менеджера файлов и обозревателя Internet (в частности, типы файлов, прокси-серверы, шифрование и панели инструментов)
	Параметры проверки орфографии...		Настроить параметры проверки орфографии в Konqueror
	Меню Окно (рис. 4.19)		
Окно	Разделить панель по вертикали	<Ctrl+Shift+L>	Разделить текущее окно по вертикали
	Разделить панель по горизонтали	<Ctrl+Shift+T>	Разделить текущее окно по горизонтали
	Удалить активную панель	<Ctrl+Shift+R>	Удалить активную панель
	Новая вкладка	<Ctrl+Shift+N>	Открыть новую вкладку
	Создать копию вкладки	<Ctrl+Shift+D>	Открыть текущую страницу в новой вкладке
	Отделить вкладку	<Ctrl+Shift+B>	Показать текущую вкладку в отдельном окне
	Закрыть вкладку	<Ctrl+W>	Закрыть текущую вкладку
	Переместить вкладку влево	<Ctrl+Shift+Left>	Переместить текущую вкладку влево

Окончание прил. 2

	Пункт	Клавиши	Действия
Окно	Переместить вкладку вправо	<Ctrl+Shift+Left>	Переместить текущую вкладку вправо
	Показать эмулятор терминала		Открыть в нижней части окна небольшое окно терминала
	Показать панель навигации	<F9>	Скрыть или показать панель навигации Konqueror
Меню Справка			
Справка	Руководство по Konqueror		Открыть руководство по Konqueror
	Что это?	<Shift+F1>	При выборе этого пункта меню указатель мыши примет форму стрелки с вопросительным знаком. Если щелкнуть мышью по какому-нибудь элементу окна, например по кнопке Остановить, будет показана краткая справка по этому элементу
	Введение в Konqueror		Открыть страницу, появляющуюся при первом запуске Konqueror и описывающую его основные возможности
	Сообщить об ошибке...		Данный пункт позволяет сообщить разработчикам об ошибке в Konqueror
	О программе Konqueror...		Показать окно с краткими сведениями о версии Konqueror, его авторах и лицензии
	О KDE...		Показать окно с краткими сведениями о версии KDE
Специальные клавиши (клавиши, которые не дублируются отдельным пунктом меню)			
Специальные	Нет	<F6>	Перейти к строке адреса
	—	<Ctrl+]>	Перейти к следующей вкладке
	—	<Ctrl+[>	Перейти к предыдущей вкладке

Примечания:

1. Приведенные сочетания клавиш являются стандартными, принятыми по умолчанию, однако их можно легко изменить.

2. Некоторые из пунктов меню появляются только при открытии соответствующих типов файлов, например, Правка\Найти... не показывается при просмотре каталога.

Оглавление

Предисловие	3
Глава 1. ОПЕРАЦИОННЫЕ СИСТЕМЫ ЭВМ. ОСНОВНЫЕ ПРИНЦИПЫ И ПОНЯТИЯ	9
1.1. Классы, функции и состав операционных систем	9
Функции ОС	9
Программы ОС	12
1.2. Управление данными в операционных системах	14
Внешние устройства ЭВМ	15
Накопители на магнитных носителях, файлы, циклы обработки	15
Адресация, имена, спецификация данных в ОС	18
Накопители на магнитных лентах	21
Накопители на магнитных дисках	25
Особенности и характеристики НМД для персональных компьютеров	27
Файловые системы	29
Разделение доступа к данным в ОС	37
Форматы файлов	38
1.3. Управление заданиями (процессами, задачами)	40
Классификация процессов	40
Классификация ресурсов	42
Управление процессами	43

Планирование процессов	44
Взаимодействие процессов	46
Планирование работы процессора	49
Стратегии планирования процессора	50
Управление неvirtуальной памятью	55
Страничная организация памяти	61
Управление виртуальной памятью	64
Алгоритм распределения страничных рамок	66
1.4. Связь с оператором	68
Связь с пользователем	68
Разновидности интерфейсов	69
Терминалы	74
Экран	79
Основные элементы графических интерфейсов (виджеты, widgets)	86
 Глава 2. ОПЕРАЦИОННЫЕ СИСТЕМЫ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ — ОДНОПОЛЬЗОВАТЕЛЬСКИЕ, ОДНОЗАДАЧНЫЕ И МНОГОЗАДАЧНЫЕ	 90
2.1. Операционная система MS DOS	90
Краткая история операционной системы MS DOS	90
Некоторые основные понятия, связанные с функционированием MS DOS	95
Основные составные части MS DOS	102
Загрузка и инициализация DOS	109
Файловые системы MS DOS	110
Управление оперативной памятью	119
Драйверы MS DOS	121
 2.2. Графические программные оболочки Windows 3.x	 122
Операционная оболочка Windows 3.1	122
Окна в Windows	124
Пиктограммы	126
Составные части окна	128
Меню	130

Диспетчер Программ	132
Диспетчер файлов	135
Приложения Windows 3.1	137
Помощь (справочная подсистема)	138
Windows for Workgroups 3.11	139
Краткие сведения об архитектуре Windows 3.x	141
2.3. Операционная система OS/2	143
Общие характеристики OS/2	143
Внутренняя организация OS/2 Warp	145
Файловая система HPFS	148
2.4. Операционные системы Windows 95/98/ME	155
Объектно-ориентированный подход	156
Windows 95/98, основные особенности	157
Основные отличия Windows 98	162
Функции и состав ОС Windows 95/98	165
Интерфейс Windows 95/98	168
Краткие сведения по архитектуре Windows 95/98 . .	188
Windows ME	191
2.5. Операционные системы Windows NT/2000/ XP/VISTA/W7	193
Windows NT	193
Архитектурные модули Windows NT	198
Основные отличия Windows 2000	208
Основные особенности Windows XP	218
Windows Vista	230
Windows 7 (Vienna)	240
Файловые системы NTFS4 (Windows NT) и NTFS5 (Windows 2000)	245
2.6. Операционные системы других аппаратных платформ	256
Windows Mobile (WM)	256
MacOS X	262

Глава 3. ОПЕРАЦИОННЫЕ СИСТЕМЫ КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ — МНОГОПОЛЬЗОВАТЕЛЬСКИЕ МНОГОЗАДАЧНЫЕ	277
3.1. Операционные системы OS/360/370/375	278
Вычислительные машины ряда ЭВМ IBM/360 (ЕС ЭВМ)	278
Основные сведения о функционировании ОС	279
Язык управления заданиями	280
Утилиты ОС IBM/360	286
3.2. Операционные системы RSX (ОС РВ)	290
Некоторые основные понятия, связанные с функционированием ОСРВ (RSX)	291
Текстовые редакторы ОС РВ	296
3.3. Операционная система Unix	297
Основные компоненты ОС Unix	298
Основные понятия, связанные с работой пользователя в ОС Unix	299
Каталоги и файлы	301
Владелец файла и защита файла	310
Работа с текстовыми файлами	311
Связь пользователь-пользователь	321
Стандартные файлы	325
Средства разработки программ	329
Системное администрирование	335
Файловые системы	343
Работа с руководствами для пользователя	352
Internet-возможности Unix	354
Управление устройствами	357
Управление процессами и потоками (threads)	359
3.4. Операционная система Linux	361
Системные характеристики	361
Оконная система X (X window system)	364

Интерактивная среда K Desktop Environment (KDE)	372
Элементы рабочего стола KDE3	373
Работа с окнами	379
Настройки интерфейса	382
Некоторые приложения KDE	387
Средства помощи	398
KDE 4	401
Взаимодействие Windows и Linux	403
GNOME (GNU Network Object Model Environment)	406
 Глава 4. СРЕДЫ И ОБОЛОЧКИ ОПЕРАЦИОННЫХ СИСТЕМ	 411
4.1. Диалоговые мониторы ЕС ЭВМ	411
Основные группы функций	412
4.2. Монитор PCTools для ПЭВМ	414
Запуск PCTools	414
Файловые функции	417
Дисковые функции	417
4.3. Оболочка NORTON COMMANDER (DOS) и ее графические аналоги для Windows	417
Основные возможности оболочки	418
Операции над файлами	421
Другие программные оболочки	425
4.4. FAR Manager — текстовая оболочка для Windows 95/98/NT/2000/XP	428
Параметры командной строки	429
Некоторые общие понятия и операции	429
Панели	432
Настройка некоторых системных параметров	437
Меню файлов	444
Меню команд	445
Прочие инструментальные возможности	452

Встроенная программа просмотра	453
Встроенный редактор	455
Помощь	458
4.5. Программная оболочка Dosshell	459
Исходный кадр оболочки MS DOS Shell	459
Операции с файлами	461
Операции с деревом каталогов	464
Запуск программ	466
Использование интерактивного справочника MS DOS Shell	468
4.6. Менеджер файлов и Web-обозреватель	
Konqueror	469
Запуск Konqueror	470
Основные элементы интерфейса Konqueror	471
Konqueror как менеджер файлов	478
Konqueror в режиме Web-обозревателя	491
Сохранение настроек и профилей	499
4.7. Некоторые другие приложения, расширяющие функции операционных систем	502
Трехмерные менеджеры рабочего стола	502
Эмуляторы DOS	506
Microsoft Virtual PC 2007	508
Wine	516
Заключение	520
Литература	523
Глоссарий (список используемых терминов)	525
Приложение 1. Команды и функции FAR MANAGER	540
Приложение 2. Главное меню Konqueror	546

Партыка Татьяна Леонидовна
Попов Игорь Иванович

Операционные системы, среды и оболочки

Учебное издание

Редактор *А. В. Волковицкая*
Корректор *А. В. Алешина*
Компьютерная верстка *И.В. Кондратьевой*
Оформление серии *П. Родькина*

Подписано в печать 25.11.2013. Формат 60×90/16.
Гарнитура «Таймс». Усл. печ. л. 35,0. Уч.-изд. л. 35,6.
Печать офсетная. Бумага офсетная. Доп. тираж 1000 экз.
Заказ №

Издательство «ФОРУМ»
101990, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (495) 625-32-07, 625-52-43
E-mail: forum-knigi@mail.ru

Отдел продаж издательства «ФОРУМ»:

101990, Москва — Центр, Колпачный пер., д. 9а
Тел./факс: (495) 625-52-43
E-mail: forum-ir@mail.ru
www.forum-books.ru

*Книги издательства «ФОРУМ»
вы также можете приобрести:*

Отдел продаж «ИНФРА-М»
127282, Москва, ул. Полярная, д. 31в
Тел.: (495) 380-05-40 (доб. 252)
Факс: (495) 363-92-12

Отдел «Книга-почтой»
E-mail: podpiska@infra-m.ru;
books@infra-m.ru