

С. В. СИНИЦЫН, А. В. БАТАЕВ, Н. Ю. НАЛЮТИН

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Учебник

Рекомендовано

*Учебно-методическим объединением по образованию
в области прикладной информатики в качестве учебника
для студентов высших учебных заведений, обучающихся
по специальности «Прикладная информатика (по областям)»
и другим экономическим специальностям*



Москва
Издательский центр «Академия»
2010

УДК 681.3.066(075.8)
ББК 32.973-018.2я73
С384

Рецензенты:

проф. кафедры математического обеспечения и администрирования информационных систем Московского государственного университета экономики, статистики и информатики канд. экон. наук, доцент *В. П. Грибанов*;
проф. кафедры управления и информатики Московского энергетического института (технического университета) д-р техн. наук *О. С. Колосов*

Синицын С. В.

С384 **Операционные системы : учебник для студ. высш. учеб. заведений / С. В. Синицын, А. В. Батаев, Н. Ю. Налютин. — М. : Издательский центр «Академия», 2010. — 304 с. ISBN 978-5-7695-6672-1**

Изложены основные принципы организации современных операционных систем (ОС) на примере ОС UNIX и Windows. Рассмотрены методы и языковые средства для работы с основными объектами, находящимися под управлением ОС: файлами, заданиями, пользователями, процессами. Значительное внимание уделено вопросам обеспечения межпроцессного взаимодействия. Текст иллюстрируется многочисленными примерами, содержит контрольные вопросы и задания.

Для студентов учреждений высшего профессионального образования.

УДК 681.3.066(075.8)
ББК 32.973-018.2я73

Оригинал-макет данного издания является собственностью Издательского центра «Академия», и его воспроизведение любым способом без согласия правообладателя запрещается

© Синицын С. В., Батаев А. В., Налютин Н. Ю., 2010
© Образовательно-издательский центр «Академия», 2010
© Оформление. Издательский центр «Академия», 2010
ISBN 978-5-7695-6672-1

ВВЕДЕНИЕ

Операционная система (ОС) — программный комплекс, предоставляющий пользователю среду для выполнения прикладных программ и управления ими, а прикладным программам средства доступа и управления аппаратными ресурсами.

Каждый пользователь ОС использует в своей деятельности инструменты, предоставляемые либо непосредственно ядром ОС, либо работающими под управлением ОС прикладными программами. Для решения своих задач пользователь формализует описание задачи на некотором входном языке для ОС или программ.

Синтаксис и семантика таких языков различна в зависимости от решаемых при их помощи задач. Эти задачи могут быть разделены на следующие группы:

- расширение функциональности ОС;
- конфигурирование режимов работы ОС;
- разработка прикладных программ;
- решение прикладных задач при помощи готовых программ.

Пользователей ОС, применяющих тот или иной язык общения с ОС, можно подразделить на несколько групп (рис. В1): системные программисты; системные администраторы; прикладные программисты; прикладные пользователи.

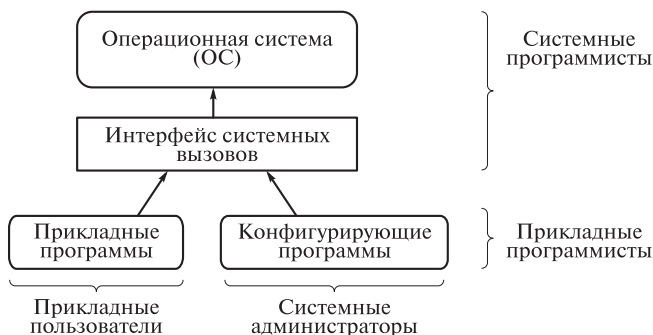


Рис. В1. Группы пользователей ОС

Один и тот же язык может служить для решения различных задач. Если более подробно классифицировать пользователей по используемому ими языку и решаемым задачам, то каждая пара «входной язык — решаемая задача» будет определять роль пользователя при его работе с ОС. В табл. В1 приведены основные типы ролей пользователей ОС.

Таблица В1. Роли пользователей ОС

Роль пользователя	Решаемая задача	Входной язык
Системный программист	Расширение функций ОС	Низкоуровневые языки разработки, в том числе Ассемблер
Системный администратор	Конфигурирование ОС и регистрация пользователей	Форматы конфигурационных файлов и языки управления средствами администрирования
Оператор	Текущее администрирование системы, установка/удаление программного обеспечения, его настройка	Форматы конфигурационных файлов инсталляторов и устанавливаемого программного обеспечения
Специалист по аппаратному обеспечению	Обслуживание аппаратуры, ввод ее в эксплуатацию, вывод из эксплуатации	Языки средств настройки оборудования для использования в конкретной ОС
Прикладной программист	Разработка программного обеспечения, предназначенного для решения задач прикладного пользователя	Языки высокого уровня, интерфейс системных вызовов ядра ОС
Администратор данных	Архивирование данных системы, управление информационными ресурсами (базы данных, справочники)	Языки управления и конфигурирования используемых программных средств
Прикладной пользователь	Решение конкретных прикладных задач при помощи готового программного обеспечения	Языки управления заданиями ОС, языки управления используемыми программными средствами

Данное учебное издание в первую очередь ориентировано на студентов, выступающих в роли прикладных программистов и пользователей ОС, но в некоторой мере затрагивает также вопросы администрирования и конфигурирования ОС.

Большая часть материала излагается на примере общих механизмов, существующих в ОС семейств UNIX и Windows. Часть материала применима и для ОС семейства DOS (в частности, раздел, посвященный заданиям в Windows).

Причина того, что речь идет о семействе ОС UNIX, а не об одной ОС, заключается в том, что семейство ОС UNIX развивалось значительное время в различных, зачастую не связанных друг с другом, коллективах разработчиков.

Первый вариант ОС UNIX был создан в 1969 г. несколькими программистами лаборатории Bell Labs фирмы AT&T и работал на компьютере PDP-7. Она использовалась для решения практических задач сотрудников лаборатории, и ее широкое распространение не планировалось. Через некоторое время большая часть ОС была переписана с языка Ассемблер на язык C, что дало возможность перенести ее на большое количество разных платформ. В настоящее время ОС Unix работает на большинстве существующих архитектур, и для многих из них является основной ОС.

Одна из следующих версий UNIX-систем, разработанных фирмой AT&T и их производных, называется System V (пятая версия), сокращенно SysV (иногда используют название «AT&T-версия UNIX»).

В середине 1970-х годов в университете Беркли была создана своя версия UNIX, получившая название BSD UNIX (Berkeley Software Distribution).

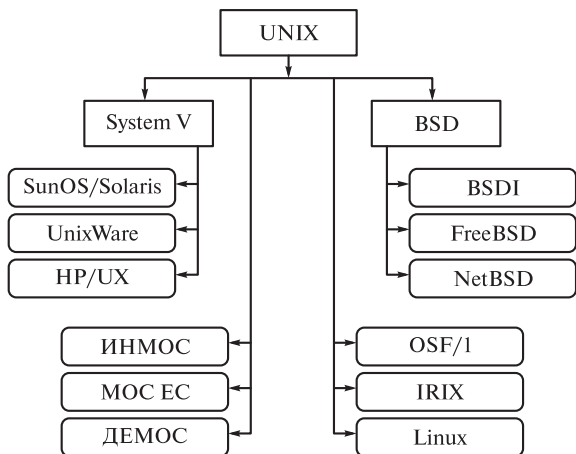


Рис. В2. Генеалогическое древо различных вариантов UNIX

Большинство вариантов ОС UNIX основаны или на System V, или на BSD (рис. В2).

Обе ветви UNIX-систем в той или иной степени удовлетворяют различным стандартам, в частности стандарту POSIX, и в настоящее время вырабатываются единые стандарты. Наиболее современные варианты UNIX, удовлетворяющие требованиям этих стандартов, нельзя четко отнести ни к той, ни к другой ветви. В их число входят IRIX (разработка Silicon Graphics), Digital OSF/1 (разработка DEC) и Linux, ранние версии которой были основаны на MINIX, разработанной Энди Таненбаумом.

Кроме того, к общему генеалогическому древу ОС семейства UNIX следует отнести отечественные разработки: ОС МОС ЕС для ЕС ЭВМ, ДЕМОС для ЕС ЭВМ и СМ ЭВМ, ИНМОС для СМ ЭВМ [6].

В качестве основной ОС рассматривается ОС Linux и Windows XP, однако большая часть материала книги применима для всех основных версий UNIX-систем и Windows семейства NT.

Учебник состоит из 9 глав и приложения:

- в главе 1 даны определения основных понятий и изложены принципы работы ОС, на которые опирается дальнейшее изложение;

- главы 2—6 дают читателю представление об основных объектах, поддерживаемых операционной системой: файлах, пользователях, заданиях и методах работы с ними;

- главы 6—9 углубляют знания читателя о файлах и пользователях. Также в этих главах вводятся в рассмотрение вопросы управления процессами и механизмы взаимодействия между процессами, поддерживаемые операционными системами UNIX и Windows;

- приложения содержат справочную информацию, а также полные исходные тексты и описание структуры данных системы «Контроль знаний», различные фрагменты которой служат примерами в первой части учебника.

Каждая глава учебника завершается контрольными вопросами и заданиями. Контрольные вопросы разделяются на две части — базовые вопросы, направленные на лучшее понимание и закрепление материала раздела, и задания, направленные на выработку практических навыков работы с UNIX- и Windows-системами.

Для понятного изложения материала большая часть приводимых примеров посвящена разработке программного комплекса, автоматизирующего процесс выдачи контрольных работ студентам и сбор выполненных контрольных работ преподавателем. При этом некоторые аспекты взаимодействия преподавателя со студентами намеренно упрощены в соответствии с изложением материала. Явно будут выделены только те моменты, которые хоро-

шо иллюстрируют различные механизмы, предоставляемые ОС пользователю.

В основной части учебника текст, относящийся к описанию программного комплекса «Контроль знаний», будет выделен шрифтом.

Программный комплекс «Контроль знаний» автоматизирует процесс выдачи преподавателем контрольных работ студентам и процесс возврата преподавателю выполненных контрольных работ (рис. В3).

Приложение «Контроль знаний» (см. приложения 1, 2) предназначено для трех основных типов пользователей:

- прикладной программист — выполняет разработку прикладной системы, расширяет ее функциональность;
- преподаватель — поддерживает базу вариантов контрольных работ по различным темам, выдает варианты работ студентам, собирает выполненные работы и проверяет их. Разбор работ и сообщение студентам оценок вынесено за рамки системы;
- студент — просматривает список полученных вариантов контрольных работ, выполняет их и возвращает выполненные работы преподавателю.

Программный комплекс «Контроль знаний» должен предоставлять средства для хранения базы вариантов контрольных работ, сгруппированных по темам. Каждая тема должна иметь уникальный номер, каждый вариант — номер, уникальный в пределах темы.

Для работы студентов им выделяется рабочая область, в которую преподавателем помещаются варианты контрольных работ, предназначенных для выполнения студентом. Внутри рабочей области студента должна быть выделена отдельная область, в которую он помещает выполненные работы. Именно из этой отдельной области преподаватель забирает работы на проверку, помещая их в свою рабочую область.

Таким образом, можно определить основные объекты, которые хранятся в программном комплексе «Контроль знаний», и действия над ними, выполняемые пользователями системы:

- база контрольных работ — основное хранилище информации о доступных вариантах. Основной объект данных — вариант. Варианты сгруппированы по темам, темы составляют общую базу;
- рабочая область студента — хранилище вариантов контрольных работ, выданных студенту. Каждый студент имеет свою собственную рабочую область. Основной объект данных — вариант контрольной работы;
- область готовых работ студента — хранилище выполненных студентом контрольных работ, готовых для проверки;
- рабочая область преподавателя — хранилище контрольных работ, собранных у студентов;

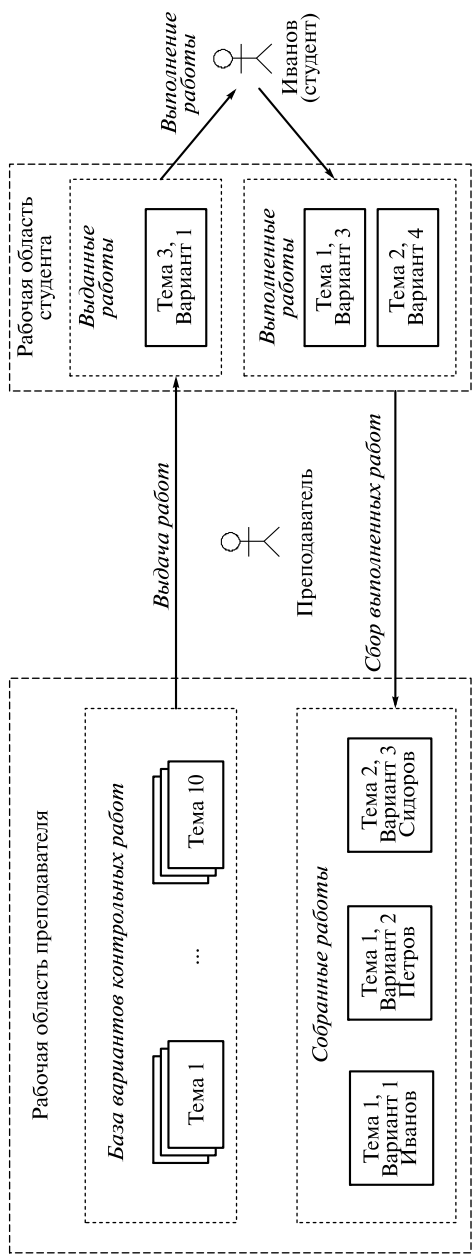


Рис. В3. Схема работы программного комплекса «Контроль знаний»

- вариант контрольной работы — список вопросов и полей, предназначенных для записи ответов студентом;

- выполненный вариант контрольной работы — вариант контрольной работы, в котором студент заполнил поля для ответов.

Программный комплекс должен автоматизировать следующие основные действия пользователя:

преподавателя:

- просмотр количества вариантов по определенной теме,
- выдачу одного варианта по теме конкретному студенту,
- выдачу вариантов по заданной теме всем студентам,
- сбор выполненных работ в свою рабочую область;

студента:

- просмотр полученных вариантов контрольных работ,
- выполнение контрольной работы,
- сдачу контрольной работы преподавателю.

При написании программного комплекса «Контроль знаний» должны учитываться следующие моменты:

- размещение и структуризация информации на дисковых носителях;

- определение прав доступа к различной информации;
- средства автоматизации типичных действий пользователя.

Прочитав внимательно учебник, вдумчивый читатель получит базовые знания о структуре ОС, навыки работы с ОС UNIX и Windows, а также общие представления о вопросах работы распределенных систем (пока только на примере межпроцессного взаимодействия). Однако круг вопросов, связанных с построением, администрированием и использованием операционных систем, гораздо шире.

В частности, в силу ограниченности объема не рассмотрены следующие вопросы:

- генерация ОС (конфигурирование ядра, администрирование) [5];

- графические интерфейсы (X Window System) [2];

- прикладные пакеты программ;

- специфика организации гетерогенных систем и сред [13];

- аспекты разработки дополнительных компонент ядра, драйверов устройств [7];

- программирование в операционных системах реального времени [14];

- микроядерные архитектуры [11];

- разработка сетевых приложений на основе различных протоколов связи [10].

Все эти вопросы заинтересованные читатели могут изучить самостоятельно.

ОБЩАЯ ХАРАКТЕРИСТИКА ОПЕРАЦИОННЫХ СИСТЕМ

1.1. Основные понятия

При решении задач в среде операционной системы (ОС) пользователь должен определить данные и инструментальное (программное) средство для их обработки. В большинстве случаев решение задачи пользователя сводится к последовательному применению нескольких инструментов (например, для ввода данных, сортировки, слияния, вывода).

Операционная система предоставляет пользователю базовый набор инструментов и среду для хранения данных, а также средства задания последовательности использования инструментов. Время, в течение которого пользователь решает последовательно одну или несколько задач, пользуясь при этом средствами, предоставляемыми ОС, называется *сеансом*. В начале любого сеанса пользователь идентифицирует себя, а в конце указывает на необходимость завершения сеанса. Последовательность использования инструментов, записанная на некотором формальном языке, называется *заданием*, сам язык — *языком управления заданиями*.

Выполнение заданий в большинстве ОС производится *командным интерпретатором*, более подробное определение которого будет дано в гл. 3. Обычно пользователю предоставляется некоторый интерфейс общения с командным интерпретатором, команды которого вводятся с клавиатуры, а результат их выполнения выводится на экран. Такой интерфейс ассоциируется с логическим понятием *терминала* — совокупности устройства ввода (обычно клавиатуры) и устройства вывода (дисплея, выводящего текстовую информацию). В настоящее время наиболее употребительным является графический интерфейс пользователя (GUI), рассмотрение которого выходит за рамки данной книги [2].

Операционная система выполняет функции управления аппаратными ресурсами, их распределения между выполняемыми программами пользователя и формирует некоторую среду, содержащую данные, необходимые для выполнения программ. Такая среда в дальнейшем будет называться *информационным окруже-*

нием. В информационное окружение входят все данные и объекты, обрабатываемые ОС, которые оказывают существенное влияние на выполнение программы. Далее будут приведены примеры информационного окружения различного характера.

Программа (в общем случае) — набор инструкций процессора, хранящийся на диске (или другом носителе информации). Чтобы программа могла быть запущена на выполнение, ОС должна создать среду выполнения — информационное окружение, необходимое для выполнения программы. После этого ОС перемещает исполняемый код и данные программы в оперативную память и инициирует выполнение программы.

Используя понятия программы, данных и информационного окружения, можно определить задачу как совокупность программ и данных, являющихся частью информационного окружения.

Выполняемая программа образует процесс. *Процесс* — совокупность информационного окружения и области памяти, содержащей исполняемый код и данные программы. Обычно в памяти, адресуемой ОС, одновременно может работать большое число процессов. Естественно, что на однопроцессорных компьютерах возможно одновременное выполнение программного кода только одного процесса, поэтому часть процессов находится в режиме ожидания, а один из процессов — в режиме выполнения. Процессы при этом образуют очередь, ОС передает управление первому процессу в очереди, затем следующему и т. д.

Процесс, имеющий потенциальную возможность получить входные данные от пользователя с клавиатуры и вывести результаты своей работы на экран, называется *процессом переднего плана*, процесс, выполняемый без непосредственного взаимодействия с пользователем, — *фоновым процессом*.

В ходе работы процессы используют вычислительную мощность процессора, оперативную память, обращаются к внешним файлам, внутренним данным ядра ОС. Все эти объекты входят в информационное окружение процесса и называются *ресурсами*.

Ресурсом может быть как физический объект, к которому ОС предоставляет доступ — процессор, оперативная память, дисковые накопители, так и логический объект, который существует только в пределах самой ОС, например таблица выполняемых процессов или сетевых подключений. Необходимость в управлении ресурсами со стороны ОС вызвана, в первую очередь, тем, что ресурсы ограничены (по объему, времени использования, количеству обслуживаемых пользователей и т. п.). В этой ситуации ОС либо управляет лимитами ресурсов, предотвращая их исчерпание, либо предоставляет средства обработки ситуаций, связанных с исчерпанием ресурсов. Лимиты многих ресурсов, заданные в ОС по умолчанию, могут изменяться затем администратором системы

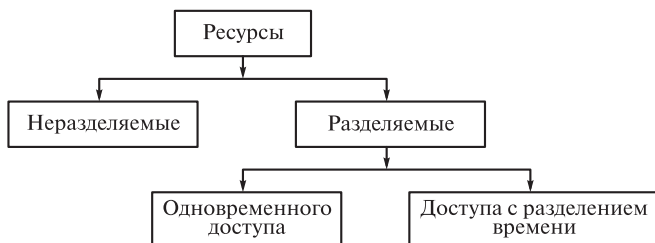


Рис. 1.1. Типы ресурсов ОС

(например, к таким ресурсам можно отнести количество файлов, одновременно открытых пользователем). В случае, если ОС позволяет одновременно использовать ресурсы нескольким процессам, ресурсы такой ОС подразделяют на типы, указанные на рис. 1.1 [1].

Неразделяемые ресурсы могут быть использованы на заданном отрезке времени только одним процессом, при этом другие процессы не имеют доступа к этому ресурсу до полного освобождения ресурса занявшим его процессом. Примером такого ресурса может служить файл, открытый на запись в исключительном режиме — все попытки использовать этот файл другими процессами (даже на чтение) завершаются неудачей.

Разделяемые ресурсы могут использоваться несколькими процессами. При этом к таким ресурсам возможен одновременный доступ процессов (например, к часам, при помощи которых определяется текущее системное время).

Некоторые разделяемые ресурсы не могут обеспечить одновременный доступ, но позволяют использовать их несколькими процессами, не дожидаясь момента полного освобождения ресурса. В этом случае используется квантование моментов использования ресурса по времени. В каждый квант времени один процесс получает полные и исключительные права на использование данного ресурса. При этом величина такого кванта заведомо много меньше полного времени, в течение которого ресурс используется одним процессом, т.е. времени, необходимого процессу для решения задачи пользователя.

Примером ресурса с доступом с разделением времени может служить процессорное время в многозадачных ОС — в каждый квант времени выполняется определенное число инструкций процесса, после чего управление передается следующему процессу и начинается выполнение его инструкций.

Процессы, ожидающие предоставления доступа к разделяемому ресурсу, организуются в очередь с приоритетом. Процессы с одинаковым приоритетом получают доступ к ресурсу последова-

тельными квантами, при этом некоторые процессы имеют более высокий приоритет и получают доступ к ресурсу чаще.

1.2. Типовая структура операционной системы

Обычно в составе ОС выделяют два уровня: ядро системы и вспомогательные системные программные средства, иногда называемые системными утилитами. Ядро выполняет все функции по управлению ресурсами системы — как физическими, так и логическими — и разделяет доступ пользователей (программ пользователей) к этим ресурсам. При помощи системного программного обеспечения пользователь управляет средствами, предоставляемыми ядром.

В ядро типичной ОС входят следующие компоненты: система управления сеансами пользователей, система управления задачами (процессами), файловая система, система ввода/вывода. Интерфейс ядра ОС с прикладными программами осуществляется при помощи программного интерфейса системных вызовов, интерфейс с аппаратным обеспечением — при помощи драйверов (рис. 1.2).

Система управления сеансами пользователей осуществляет регистрацию сеанса пользователя при начале его работы с ОС, хранит оперативную информацию, входящую в информационное окружение сеанса, при помощи системы ввода/вывода поддерживает соответствие пользовательского терминала реальным или

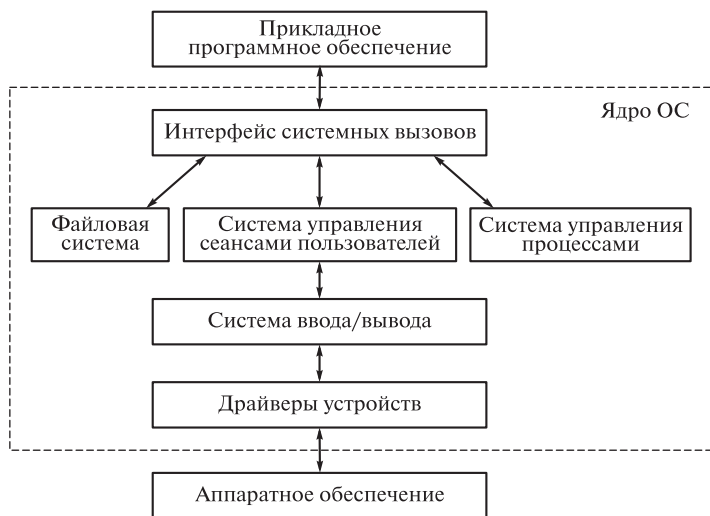


Рис. 1.2. Структура ядра типичной ОС

виртуальным устройствам, корректно завершает сеанс при окончании работы пользователя с системой.

Система управления процессами распределяет ресурсы между выполняемыми задачами (процессами), обеспечивает защиту памяти процессов от модификации ее другими процессами, реализует механизмы межпроцессного взаимодействия.

Файловая система выполняет преобразование данных, хранимых на внешних запоминающих устройствах (например, на дисковых накопителях или на flash-накопителях), в логические объекты — файлы и каталоги. Она также выполняет функции разграничения доступа к файлам и каталогам при обращении к ним со стороны системы управления сеансами или при использовании файловой системы через интерфейс системных вызовов.

Система ввода/вывода обрабатывает запросы всех рассмотренных выше компонентов ядра и преобразовывает их в вызовы логических устройств, поддерживаемых ОС. Каждое такое устройство представляет собой логический объект, обращение к которому происходит стандартными для ОС средствами (например, как к адресу в оперативной памяти либо как к специальному файлу). Логическое устройство может быть чисто виртуальным (целиком функционировать внутри ядра ОС) или представлять логический объект, связанный через драйверы с реальными аппаратными устройствами.

Примером чисто виртуального устройства может служить «черная дыра» */dev/null* в UNIX-системах. Вся информация, записываемая в это устройство, пропадает, т. е. оно может быть использовано для поглощения данных, не существенных для решаемой задачи.

Драйверы устройств преобразуют запросы системы ввода/вывода в последовательности управляющих команд для аппаратных устройств. Драйвер каждого устройства скрывает особенности его аппаратной реализации и предоставляет системе ввода/вывода стандартизированный интерфейс доступа к аппаратному обеспечению системы.

С точки зрения прикладного программиста доступ к компонентам ядра ОС осуществляется при помощи интерфейса системных вызовов — набора библиотек, включающих в себя стандартизированные наборы функций. Каждый такой набор предназначен для решения того или иного класса прикладных задач: доступа к сетевым ресурсам, графическому режиму, реализации межпроцессного взаимодействия и т. п.

1.3. Классификация операционных систем

Сложность составных частей ядра ОС и реализуемые ими функции, в первую очередь, зависят от числа одновременно обслужи-

ваемых ОС пользователей и от числа одновременно выполняемых процессов. В связи с этим разумно провести классификацию ОС по этим двум параметрам и рассмотреть особенности компонентов ядра в каждом из типов ОС.

По числу одновременно обслуживаемых пользователей ОС подразделяют на однопользовательские (одновременно поддерживается не более одного сеанса пользователя) и многопользовательские (одновременно поддерживается множество сеансов пользователя).

Многопользовательские системы, кроме обеспечения защиты данных пользователей от несанкционированного доступа других пользователей, предоставляют средства разделения общих данных между многими пользователями. Рассмотрим особенности этих типов ОС более подробно.

По числу одновременно выполняемых процессов ОС подразделяют на однозадачные (не более одного работающего процесса) и многозадачные (множество работающих процессов). Одним из основных отличий многозадачных систем от однозадачных является наличие средств управления доступом к ресурсам — разделение ресурсов и блокировки используемых ресурсов.

Однопользовательские ОС. Этот тип ОС обеспечивает одновременную поддержку только одного сеанса работы пользователя. Новый сеанс работы пользователя может быть начат только после завершения предыдущего сеанса. При этом новый сеанс пользователя имеет то же самое информационное окружение.

С точки зрения однопользовательской ОС пользователи неразличимы, поэтому если такую ОС начинают использовать несколько пользователей, то каждому из них операционная система предоставляет доступ ко всем ресурсам и, возможно, к одному и тому же информационному окружению. При этом пользователь может работать и со своими уникальными данными, например с данными на съемных дисках. При такой работе информационное окружение каждого сеанса работы пользователя различно.

Система управления сеансами однопользовательских ОС включает в себя только средства инициации и завершения сеанса и средства поддержки информационного окружения пользователя. Причем во многих однопользовательских ОС (например, DOS) момент инициации сеанса пользователя наступает сразу же после загрузки ядра и инициализационных сценариев. Момент завершения сеанса совпадает с моментом выгрузки ядра ОС из памяти (непосредственно перед или вследствие обесточивания оборудования). Таким образом, время жизни сеанса пользователя в однопользовательских ОС приблизительно равно времени жизни работающего ядра системы.

Вследствие неразличимости пользователей система управления сеансами и файловая система в значительной мере упрощаются. Система управления сеансами однопользовательских ОС не включает в себя средств идентификации и аутентификации пользователей, а также средств защиты информационного окружения их сеансов.

Файловая система однопользовательских ОС, как правило, не содержит сложных механизмов разграничения доступа к файлам и каталогам, хотя в файловой системе могут существовать флаги, задающие режимы работы с файлами и каталогами — их атрибуты.

Поддержка ОС только одного сеанса работы пользователя не исключает возможности одновременного выполнения многих задач пользователя. Иными словами, однопользовательская ОС может быть многозадачной.

Многопользовательские ОС. Этот тип ОС обеспечивает одновременную работу большого количества пользователей, что в значительной мере расширяет набор функций, реализуемых системой поддержки сеансов и файловой системой. В несколько меньшей степени поддержка множества пользователей отражается на системе ввода/вывода и системе управления процессами.

Система управления сеансами пользователей должна включать в себя средства идентификации и аутентификации пользователей, обеспечивать связывание каждого сеанса с реальным или виртуальным терминалом, содержать средства инициализации начального информационного окружения сеанса, обеспечивать защиту данных сеанса.

Файловая система многопользовательских ОС обеспечивает разграничение доступа к файлам и каталогам на основании идентификаторов пользователей, полученных от системы управления сеансами. Каждый файл и каталог в файловой системе сопровождается информационным блоком, определяющим права доступа к нему пользователей. Пользователю предоставляется возможность определять права таким образом, чтобы только он имел доступ к данным, содержащимся в файлах и каталогах, а другие пользователи не могли не только изменить эти данные, но даже прочитать их. При необходимости в совместном доступе к одной и той же информации может быть определен доступ на чтение и запись для многих пользователей.

Система ввода/вывода многопользовательских ОС, кроме непосредственного доступа к устройствам и буферизации ввода/вывода, также управляет разделением доступа пользователей к устройствам, т. е. управляет устройствами как разделяемыми ресурсами.

Следует отметить, что многопользовательские ОС обычно являются еще и многозадачными, поскольку они должны обеспечи-

вать одновременное выполнение большого количества программ различных пользователей.

Однозадачные ОС. Такие ОС предназначены для одновременного выполнения только одной задачи. Сразу после старта системы управление передается программе, играющей роль оболочки для работы пользователя. Как правило, одна из функций такой оболочки — запуск других программ.

Перед запуском программы сохраняется информационное окружение оболочки. После запуска программы ее процессу передается полное управление и предоставляется доступ ко всем ресурсам. По завершению программы освобождается память процесса, восстанавливается информационное окружение оболочки, после чего ОС берет управление на себя. Запуск программ в таких ОС чисто последовательный. В случае, если одной из программ требуется вызвать на выполнение другую программу, точно так же сохраняется окружение вызывающей программы и по завершению вызываемой программы окружение восстанавливается.

Система ввода/вывода однозадачных ОС не включает в себя средств разделения доступа к устройствам, поскольку устройство используется одновременно только одним процессом.

Однозадачные ОС могут быть и многопользовательскими. Примером таких систем являются ОС с пакетной обработкой. В таких ОС пользователи формируют очередь заданий на выполнение программ, при этом задания могут принадлежать различным пользователям. Система последовательно выполняет программы разных пользователей, перед сменой пользователя завершается сеанс работы предыдущего пользователя и начинается сеанс нового. Таким образом, при смене задания осуществляется смена информационных окружений каждой программы.

Многозадачные ОС. В многозадачных ОС в один момент времени в системе может быть запущено много программ (процессов). В этом случае система управления процессами включает в себя планировщик процессов, выполняющий следующие функции:

- создание и уничтожение процессов — загрузка программы в память, создание информационного окружения и передача управления процессу при его создании, удаление информационного окружения и выгрузка процесса из памяти при его уничтожении;
- распределение системных ресурсов между процессами — планирование выполнения процессов, формирование очереди процессов и управление приоритетами процессов в очереди;
- межпроцессное взаимодействие — распределение общих данных между процессами или пересылка управляющих воздействий между одновременно выполняемыми процессами;