

Структура операционных систем. Виды ядра операционных систем

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

Преподаватель: Сидиков И.Д.

Монолитное ядро

По сути, ОС – это программа, которую можно реализовать с использованием процедур и функций. Если при этом ОС компонуется как одна программа, работающая в привилегированном режиме и использующая быстрые переходы с одной процедуры на другую, не требующие переключения из привилегированного режима в пользовательский режим, и наоборот, то такая архитектура построения ОС называется монолитным ядром (англ. monolithic kernel).

Архитектура «монолитное ядро» характеризуется тем, что:

- Каждая процедура может вызвать каждую;
- Все процедуры работают в привилегированном режиме;
- Все части монолитного ядра работают в одном адресном пространстве;
- Ядро «совпадает» со всей ОС;
- Сборка (компиляция) ядра осуществляется отдельно для каждого компьютера, при установке, добавлении или исключении отдельных компонент требуется перекомпиляция;
- Старейший способ организации ОС.

Монолитное ядро

Архитектура «монолитное ядро» имеют долгую историю развития и усовершенствования и, на данный момент, являются наиболее архитектурно зрелыми и пригодными к эксплуатации. Вместе с тем, монолитность ядер усложняет их отладку, понимание кода ядра, добавление новых функций и возможностей, удаление кода, унаследованного от предыдущих версий. «Разбухание» кода монолитных ядер также повышает требования к объёму оперативной памяти, требуемому для функционирования ядра ОС. Это делает монолитные ядерные архитектуры мало пригодными к эксплуатации в системах, сильно ограниченных по объёму ОЗУ, например, встраиваемых системах, производственных микроконтроллерах и т.д.

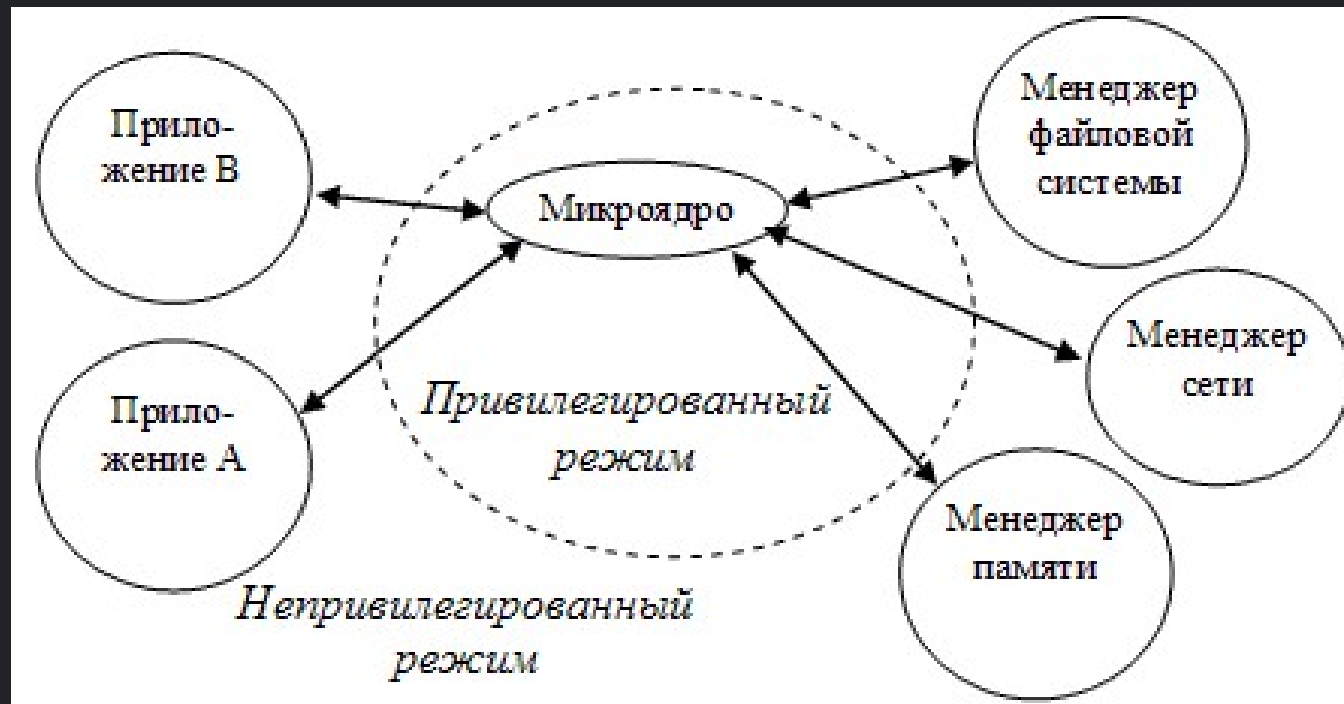
Старые монолитные ядра требовали перекомпиляции при любом изменении состава оборудования. Следует отметить, что большинство современных ядер позволяют во время работы динамически подгружать модули, выполняющие части функции ядра. Такие ядра называются модульными ядрами. Возможность динамической подгрузки модулей не нарушает монолитности архитектуры ядра, так как динамически подгружаемые модули загружаются в адресное пространство ядра и в дальнейшем работают как интегральная часть ядра. Не следует путать модульность ядра с гибридной или микроядерной архитектурой.

Примером систем с монолитным ядром служит большинство Unix-подобных систем, таких как BSD или Linux.

Микроядерная архитектура

При разработке ОС используют подход, при котором значительную часть системного кода переносят на уровень пользователя с одновременной минимизацией ядра. Системы, разработанные с использованием такого подхода, называют реализованными в микроядерной архитектуре (англ. microkernel architecture). В этом случае построение ядра ОС осуществляется так, что большинство составляющих ОС являются самостоятельными программами, а взаимодействие между ними обеспечивает специальный модуль ядра – микроядро, работающее в привилегированном режиме и обеспечивающее взаимодействие между программами, планирование использования процессора, первичную обработку прерываний, операции ввода-вывода и базовое управление памятью.

Микроядерная архитектура



Микроядерная архитектура

В микроядерных ОС выделяют центральный компактный модуль, относящийся к супервизорной части системы. Этот модуль имеет очень небольшие размеры и выполняет относительно небольшое количество управляющих функций, но позволяет передать управление на другие управляющие модули, которые и выполняют затребованную функцию. Микроядро – это минимальная главная (стержневая) часть ОС, служащая основой модульных и переносимых расширений. Микроядро само является модулем системного ПО, работающим в наиболее приоритетном состоянии компьютера и поддерживающим связи с остальной частью операционной системы, которая рассматривается как набор серверных приложений (служб).

Основная идея, заложенная в технологию микроядра, заключается в том, чтобы создать необходимую среду верхнего уровня иерархии, из которой можно легко получить доступ ко всем функциональным возможностям уровня аппаратного обеспечения. При этом микроядро является стартовой точкой для создания всех остальных модулей системы. Остальные модули, реализующие необходимые системе функции, вызываются из микроядра и выполняют сервисную роль, получая при этом статус обычного процесса.

Микроядерная архитектура

Важнейшая задача при разработке микроядра заключается в выборе базовых примитивов, которые должны находиться в микроядре для обеспечения необходимого и достаточного сервиса. В микроядре содержится и исполняется минимальное количество кода, необходимое для реализации основных системных вызовов. В число этих вызовов входят передача сообщений и организация другого общения между внешними по отношению к микроядру процессами, поддержка управления прерываниями, а также ряд других весьма немногочисленных функций. Остальные системные функции, характерные для «обычных» (не микроядерных) операционных систем, обеспечиваются как модульные дополнения-процессы, взаимодействующие главным образом между собой посредством передачи сообщений.

Микроядерная архитектура

К преимуществам построения ОС в данной архитектуре относят:

- Упрощенное добавление и отладка компонентов ОС без необходимости перезапуска системы за счет высокой степени модульности ядра;
- Возможность без прерывания работы системы, загружать и выгружать новые драйверы, файловые системы и т.Д.
- Возможность отладки компонентов ядра с помощью обычных программных средств;
- Повышенная надежность системы (ошибка на уровне непривилегированной программы менее опасна, чем отказ на уровне режима ядра).

К недостаткам построения ОС в данной архитектуре относят:

- Дополнительные накладные расходы, связанные с передачей сообщений;
- Усложнение процесса проектирования при попытке снижения возможных накладных расходов (требуется «аккуратное» проектирование, разбиение системы на компоненты, минимизация взаимодействия между ними).

Многоуровневые системы

Обеспечивая строгую структуризацию, можно представить всю вычислительную систему в виде ряда уровней с хорошо определенными связями между ними. При этом объекты уровня N могут вызывать только объекты уровня $N-1$. Чем ниже уровень, тем более привилегированные команды и действия может выполнять модуль, находящийся на этом уровне. Впервые такой подход был применен при создании системы THE (Technische Hogeschool Eindhoven) в 1968 г. Дейкстрой (Dijkstra) и его студентами

5	Интерфейс пользователя
4	Управление вводом-выводом
3	Драйвер устройства связи оператора и консоли
2	Планирование задач и процессов
1	Управление памятью
0	Аппаратное обеспечение

Многоуровневые системы

Вычислительные системы, реализованные в подобной архитектуре, называют многоуровневыми системами (англ. layered systems).

В качестве достоинства многоуровневых систем отмечают:

- Простоту реализации (за счет того, что при использовании операций нижнего слоя не нужно знать, как они реализованы, нужно лишь понимать, что они делают);
- Простоту тестирования (отладка осуществляется послойно и при возникновении ошибки всегда легко локализовать ошибку);
- Простоту модификации (при необходимости можно заменить лишь один слой, не трогая остальные).

К недостаткам относят:

- Сложность разработки (непросто верно определить порядок и состав каждого из слоев);
- Меньшая по сравнению с монолитными системами эффективность за счет необходимости прохождения целого ряда слоев (например, для выполнения операций ввода-вывода программе пользователя придется последовательно проходить все слои от верхнего до нижнего).

Виртуальные машины

Виртуальной машиной (англ. virtual machine) называют программную или аппаратную среду, исполняющую некоторый код (например, байт-код или машинный код реального процессора). Зачастую виртуальная машина эмулирует работу реального компьютера. На виртуальную машину, так же как и на реальный компьютер можно установить ОС, у виртуальной машины может быть BIOS, оперативная память, жёсткий диск (выделенное место на жёстком диске реального компьютера), могут эмулироваться периферийные устройства. На одном компьютере может функционировать несколько виртуальных машин. На рисунке представлена обобщенная структура некоторой виртуальной машины с тремя различными ОС.

Виртуальная машина реализует для пользователя имитацию hardware в вычислительной системе (процессор, привилегированные и непривилегированные команды, устройства ввода-вывода, прерывания и т.д.). При обращении к «виртуальному hardware» на уровне привилегированных команд в действительности происходит системный вызов реальной ОС, которая и производит все необходимые действия.

Виртуальные машины

Недостатками реализации ОС в подобных архитектурах является снижение эффективности виртуальных машин по сравнению с реальным компьютером, и, как правило, их громоздкость. Преимуществом является использование в рамках одной вычислительной системы программ, созданных для разных ОС. Примерами ОС, реализованных в подобной архитектуре, являются CP/CMS (VM/370) для семейства машин IBM/370, VMWare Workstation компании VMWare.

Программа пользователя	Программа пользователя	Программа пользователя
<i>MS-DOS</i>	<i>Linux</i>	<i>Windows NT</i>
Виртуальное hardware	Виртуальное hardware	Виртуальное hardware
Реальная операционная система		
Реальное аппаратное обеспечение		

Смешанные системы

В большинстве случаев современные ОС используют различные комбинации подходов, рассмотренных в пп. 2.1.1-2.1.4, реализуя смешанные (гибридные) ОС. Например, ядро ОС Linux представляет собой монолитную систему с элементами микроядерной архитектуры. Системы 4.4BSD и MkLinux – монолитные ОС, работающие на микроядре Mach (микроядро обеспечивает управление виртуальной памятью и работу низкоуровневых драйверов; остальные функции, в том числе взаимодействие с прикладными программами, осуществляется монолитным ядром). Совместно элементы микроядерной архитектуры и элементы монолитного ядра используются в ядре Windows NT:

- Компоненты ядра windows NT располагаются в вытесняемой памяти и взаимодействуют друг с другом путем передачи сообщений, как и положено в микроядерных ОС;
- Все компоненты ядра работают в одном адресном пространстве и активно используют общие структуры данных.

Классификация ОС

В зависимости от выбранного признака, по которому один объект отличают от другого, вариантов классификации может быть множество. Что касается ОС, здесь уже давно сформировалось относительно небольшое количество классификаций: по назначению, по режиму обработки задач, по способу взаимодействия с системой и, наконец, по способам построения (архитектурным особенностям системы).

Прежде всего, традиционно различают ОС общего и специального назначения. Системы специального назначения, в свою очередь, подразделяются на ОС для носимых микрокомпьютеров и различных встроенных систем, организации и ведения баз данных, решения задач реального времени и т.п. Еще недавно ОС для персональных компьютеров относили к ОС специального назначения. Сегодня современные мультизадачные ОС для персональных компьютеров уже многими относятся к ОС общего назначения, поскольку их можно использовать для самых разнообразных целей.

Классификация ОС

По режиму обработки задач различают ОС, обеспечивающие однопрограммный и мультипрограммный (мультизадачный, многозадачный) режимы. Любая задержка в решении программы (например, для осуществления операций ввода-вывода данных) используется для выполнения других программ. Однозадачные ОС (например, MS-DOS, MSX) выполняют функцию предоставления пользователю виртуальной машины, делая более простым и удобным процесс взаимодействия пользователя с компьютером, а также включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.

Следует различать понятия «мультипрограммный режим» и «мультизадачный режим». Принципиальное отличие этих понятий заключается в том, что мультипрограммный режим обеспечивает параллельное выполнение нескольких приложений, и при этом программисты, создающие эти программы, не должны заботиться о механизмах организации их параллельной работы (эти функции берет на себя сама ОС). Мультизадачный режим, наоборот, предполагает, что забота о параллельном выполнении и взаимодействии приложений ложится как раз на прикладных программистов. Современные ОС для персональных компьютеров реализуют мультипрограммный и мультизадачный режимы работы.

Классификация ОС

Среди множества существующих вариантов реализации многозадачности можно выделить две группы:

1. Не вытесняющая многозадачность (NetWare, Windows 3.x) – механизм планирования процессов целиком сосредоточен в ОС. В этом случае активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление ОС для того, чтобы та выбрала из очереди другой готовый к выполнению процесс.
2. Вытесняющая многозадачность (Windows NT, OS/2, Unix) – механизм планирования процессов распределен между системой и прикладными программами. При вытесняющей многозадачности решение о переключении процессора с одного процесса на другой принимается ОС, а не самим активным процессом.

Классификация ОС

Также многозадачные ОС подразделяют на различные типы в соответствии с использованными при их разработке критериями эффективности:

- Системы пакетной обработки (например, ЕС, критерий – коэффициент загрузки процессора);
- Системы разделения времени (unix, VMS, критерий – удобство и эффективность работы пользователей при одновременном выполнении нескольких пользовательских приложений);
- Системы реального времени (QNX, RT/11, критерий – реактивность).

Классификация ОС

При организации работы с вычислительной системой в диалоговом режиме можно говорить об однопользовательских (однотерминальных) и многопользовательских (мультитерминальных) ОС. В мультитерминальных ОС с одной вычислительной системой одновременно могут работать несколько пользователей, каждый со своего терминала. При этом у пользователей возникает иллюзия, что у каждого из них имеется собственная вычислительная система. Очевидно, что для организации мультитерминального доступа к вычислительной системе необходимо обеспечить мультипрограммный режим работы. В качестве одного из примеров мультитерминальных ОС для персональных компьютеров можно назвать Linux. Некая имитация мультитерминальных возможностей имеется и в системе Windows XP. В этой ОС каждый пользователь после регистрации (входа в систему) получает свою виртуальную машину.

Если необходимо временно предоставить компьютер другому пользователю, вычислительные процессы первого можно не завершать, а просто для этого другого пользователя система создает новую виртуальную машину. В результате компьютер будет выполнять задачи и первого, и второго пользователя. Количество параллельно работающих виртуальных машин определяется имеющимися ресурсами. Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей.

Классификация ОС

Кроме того, если в ОС отсутствуют или присутствуют средства поддержки многопроцессорной обработки, они могут быть разделены на многопроцессорные и однопроцессорные. Как правило, функции мультипроцессирования имеются в операционных системах Solaris 2.x фирмы Sun, Open Server 3.x компании Santa Crus Operations, OS/2 фирмы IBM, Windows NT фирмы Microsoft, NetWare 4.1 фирмы Novell, однако, очевидно, их наличие усложняет алгоритмы управления ресурсами. В свою очередь, многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса в системе с многопроцессорной архитектурой: асимметричные ОС и симметричные ОС.

1. Асимметричная ОС целиком выполняется только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам.
2. Симметричная ОС полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Классификация ОС

Следует отметить еще один признак, по которому разделяют ОС – организация работы с вычислительной сетью. По этому признаку выделяют сетевые ОС и распределенные ОС (следует отметить, что иногда в литературе такое разделение отсутствует). Сетевая ОС характеризуется тем, что наделена развитыми функциями работы с сетью, а также контроля доступа к файлам (систему прав доступа). К сетевым ОС относят как системы для рабочих мест (Novell for DOS, MS Windows, GNU/Linux), так серверные ОС (GNU/Linux, семейство BSD-систем, серверные версии MS Windows), а также специализированные ОС сетевого оборудования (Cisco IOS) .

При использовании распределенной ОС пользователь не знает, на локальной или удаленной машине хранятся его файлы и выполняются его программы, он может не знать, подключен ли его компьютер к сети. Внешне распределенная ОС выглядит как обычная автономная система, а ее внутреннее строение имеет существенные отличия от автономных систем.

Также ОС классифицируют по архитектуре, в которой они реализованы.