

Операционные системы

Часть 1

Определение

Операционная система (ОС) - это программа, которая обеспечивает возможность рационального использования оборудования компьютера удобным для пользователя образом.

Структура вычислительной системы

- Из чего состоит любая вычислительная система? В первую очередь, это hardware, или техническое обеспечение: процессор, память, монитор, дисковые устройства и т.д., обычно объединенные магистральным соединением, которое называется шиной
- Во вторую очередь это программное обеспечение. Все программное обеспечение принято делить на две части: прикладное и системное.

К прикладному программному обеспечению, как правило, относятся разнообразные банковские и прочие business программы, игры, текстовые процессоры, и т.п.

Под системным программным обеспечением обычно понимают программы, способствующие функционированию и разработке прикладных программ.

Структура вычислительной системы

Деление на прикладное и системное программное обеспечение является отчасти условным и зависит от того, кто осуществляет такое деление.

Так, обычный пользователь, неискушённый в программировании, может считать Microsoft Word системной программой, а с точки зрения программиста это приложение. Компилятор языка Си для обычного программиста это системная программа, а для системного прикладная. Несмотря на эту нечеткую грань, эту ситуацию можно отобразить в виде последовательности слоев:

Структура вычислительной системы



Слои программного обеспечения компьютерной системы.

Операционная система как виртуальная машина

Архитектура большинства компьютеров на уровне машинных команд очень неудобна для ее использования прикладными программами. Например, работа с диском предполагает знакомство с внутренним устройством его электронного компонента - контроллера для ввода команд вращения диска, поиска и форматирования дорожек, чтения и записи секторов и т.д. Ясно, что средний программист не в состоянии учитывать все особенности работы оборудования (в современной терминологии заниматься разработкой драйверов устройств), а должен иметь простую высокоуровневую абстракцию, скажем, представляя информационное пространство диска как набор файлов

Файл можно открывать для чтения или записи, использовать для получения или сброса информации, а потом закрывать. Это концептуально проще, чем заботиться о деталях перемещения головок дисков или организации работы мотора. Аналогичным образом, с помощью простых и ясных абстракций, скрываются от программиста все ненужные ему подробности организации прерываний, работы таймера, управления памятью и т.д. Более того, на современных вычислительных комплексах может быть создана иллюзия неограниченного размера операционной памяти и числа процессоров. Всем этим занимается операционная система. Таким образом, операционная система представляется пользователю виртуальной машиной, с которой проще иметь дело, чем непосредственно с оборудованием компьютера.

Операционная система как менеджер ресурсов

Операционная система предназначена для управления всеми частями весьма сложной архитектуры компьютера. Представим для примера, что случится, если несколько программ, работающих на одном компьютере, будут пытаться одновременно осуществлять вывод на принтер. Мы получили бы неупорядоченную смесь строчек и страниц, выведенных различными программами.

Операционная система предотвращает хаос такого рода за счет буферизации информации, предназначенной для печати, на диске и организации очереди на печать. Для многопользовательских компьютеров, необходимость управления ресурсами и их защиты еще более очевидна.

Следовательно, операционная система как менеджер ресурсов, осуществляет упорядоченное и контролируемое распределение процессоров, памяти и других ресурсов между различными программами, их использующими.

Операционная система как защитник пользователей и программ

Если вычислительная система допускает совместную работу нескольких пользователей, то возникает проблема организации их безопасной деятельности. Необходимо обеспечить сохранность информации на диске, чтобы никто не мог удалить или повредить чужие файлы. Нельзя разрешить программам одних пользователей произвольно вмешиваться в работу программ других пользователей. Нужно пресекать попытки несанкционированного использования вычислительной системы. Всю эту деятельность осуществляет операционная система как организатор безопасной работы пользователей и их программ. С такой точки зрения операционная система выглядит системой безопасности в государстве, на которую возложены полицейские и контрразведывательные функции.

Операционная система как постоянно функционирующее ядро

Наконец, можно дать и такое определение:

операционная система это программа, постоянно работающая на компьютере и взаимодействующая со всеми прикладными программами. Казалось бы, это абсолютно правильное определение, но во многих современных операционных системах постоянно работает на компьютере лишь часть операционной системы, которую принято называть ее ядром.

Краткая история эволюции ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

- *Первый период (1945 - 1955).* Ламповые машины. Операционные систем отсутствовали.
- *Второй период (1955 - Начало 60-х).* Компьютеры на основе транзисторов. Пакетные операционные системы
- *Третий период (Начало 60-х - 1980).* Компьютеры на основе интегральных микросхем. Первые многозадачные ОС.
- *Четвертый период (1980 – настоящее время).* Персональные компьютеры. Классические, сетевые и распределенные системы.

Появление мультипрограммирования требует целой революции в строении вычислительной системы. Большую роль, здесь играет аппаратная поддержка, наиболее существенные особенности которой:

- Реализация защитных механизмов. Программы не должны иметь самостоятельного доступа к распределению ресурсов, что приводит к появлению привилегированных и непривилегированных команд. Привилегированные команды, например команды ввода-вывода, могут исполняться только операционной системой. Говорят, что она работает в привилегированном режиме. Переход управления от прикладной программы к ОС сопровождается контролируемой сменой режима. Во-вторых, это защита памяти, позволяющая изолировать конкурирующие пользовательские программы друг от друга, а ОС от программ пользователей.

Появление мультипрограммирования требует целой революции в строении вычислительной системы. Большую роль, здесь играет аппаратная поддержка, наиболее существенные особенности которой:

- Наличие прерываний. Внешние прерывания оповещают ОС о том, что произошло асинхронное событие, например, завершилась операция ввода-вывода. Внутренние прерывания (сейчас их принято называть исключительными ситуациями) возникают, когда выполнение программы привело к ситуации, требующей вмешательства ОС, например, деление на ноль или попытка нарушения защиты.

Роль операционной системы в организации мультипрограммирования

- Интерфейс между прикладной программой и ОС был организован при помощи набора системных вызовов.
- Организация очереди из заданий в памяти и выделение процессора одному из заданий потребовали планирования заданий.
- Для переключения процессора с одного задания на другое возникла потребность в сохранении содержимого регистров и структур данных, необходимых для выполнения задания, иначе говоря, контекста, для обеспечения правильного продолжения вычислений.
- Поскольку память является ограниченным ресурсом, оказались нужны стратегии управления памятью, то есть потребовалось упорядочить процессы размещения, замещения и выборки информации из памяти.
- Так как программы могут пожелать произвести санкционированный обмен данными, стало необходимо их обеспечить средствами коммуникации.
- Для корректного обмена данными необходимо предусмотреть координацию программами своих действий, т.е. средства синхронизации.

Основные функции классических операционных:

- Планирование заданий и использования процессора.
- Обеспечение программ средствами коммуникации и синхронизации.
- Управление памятью.
- Управление файловой системой.
- Управление вводом-выводом.
- Обеспечение безопасности

Каждая из приведенных функций обычно реализована в виде подсистемы, являющейся структурным компонентом ОС. В каждой конкретной операционной системе эти функции, конечно, реализовывались по-своему, в различном объеме. Они не были придуманы как составные части деятельности операционных систем изначально, а появились в процессе развития, по мере того, как вычислительные системы становились удобнее, эффективнее и безопаснее.

Основные понятия ОС.

- *Системные вызовы*
- *Прерывания*
- *Исключительные ситуации*
- *Файлы*
- *Процессы, нити*

Системные вызовы

В любой операционной системе поддерживается некоторый механизм, который позволяет пользовательским программам обращаться за услугами ядра ОС. Эти средства общения с ядром часто назывались экстракодами или системными макрокомандами. В ОС UNIX такие средства называются системными вызовами.

Системные вызовы (system calls) интерфейс между операционной системой и пользовательской программой. Они создают, удаляют и используют различные объекты, главные из которых процессы и файлы. Пользовательская программа запрашивает сервис у операционной системы, осуществляя системный вызов. Имеются библиотеки процедур, которые загружают машинные регистры определенными параметрами и осуществляют прерывание процессора, после чего управление передается обработчику данного вызова, входящему в ядро операционной системы. Цель таких библиотек сделать системный вызов похожим на обычный вызов подпрограммы.

Основное отличие состоит в том, что при системном вызове задача переходит в привилегированный режим или режим ядра (kernel mode). Поэтому системные вызовы иногда еще называют программными прерываниями в отличие от аппаратных прерываний, которые чаще называют просто прерываниями.

В этом режиме работает код ядра операционной системы, причем он выполняется в адресном пространстве и в контексте вызвавшей его задачи. Таким образом, ядро операционной системы имеет полный доступ к памяти пользовательской программы, и при системном вызове достаточно передать адреса одной или нескольких областей памяти с параметрами вызова и адреса одной или нескольких областей памяти для результатов вызова.

В большинстве операционных систем системный вызов осуществляется командой программного прерывания (INT). Таким образом, программное прерывание это синхронное событие.

Прерывания

- Прерывание (hardware interrupt) событие, генерируемое внешним (по отношению к процессору) устройством. Посредством аппаратных прерываний аппаратура либо информирует центральный процессор о том, что возникло какое-либо событие, требующее немедленной реакции (например, пользователь нажал клавишу), либо сообщает о завершении асинхронной операции ввода-вывода (например, закончено чтение данных с диска в основную память).
- Важный тип аппаратных прерываний - прерывания таймера, которые генерируются периодически через фиксированный промежуток времени. Прерывания таймера используются операционной системой при планировании процессов. Каждый тип аппаратных прерываний имеет собственный номер, однозначно определяющий источник прерывания.
- Аппаратное прерывание это асинхронное событие, то есть оно возникает вне зависимости от того, какой код исполняется процессором в данный момент. Обработка аппаратного прерывания не должна учитывать, какой процесс является текущим.

Исключительные ситуации

- Исключительная ситуация (exception) событие, возникающее в результате попытки выполнения программой недопустимой команды, доступа к ресурсу при отсутствии достаточных привилегий или обращения к отсутствующей странице памяти.
- Исключительные ситуации так же, как и системные вызовы, являются синхронными событиями, возникающими в контексте текущей задачи. Исключительные ситуации можно разделить на *исправимые* и *неисправимые*. К исправимым относятся такие исключительные ситуации, как отсутствие нужной информации в оперативной памяти. После устранения причины исправимой исключительной ситуации программа может продолжить выполнение.
- Возникновение в процессе работы операционной системы исправимых исключительных ситуаций является нормальным явлением. Неисправимые исключительные ситуации обычно возникают в результате ошибок в программах. Обычно операционная система реагирует на такие ситуации завершением программы, вызвавшей исключительную ситуацию.

Файлы

- Файлы предназначены для хранения информации на внешних носителях, то есть, принято, что информация, лежащая, например, на диске, должна находиться внутри файла. Обычно под файлом понимают часть пространства на носителе информации, имеющую имя.
- Главная задача файловой системы (file system) скрыть особенности ввода-вывода и дать программисту простую абстрактную модель файлов, независимых от устройств. Для чтения, создания, удаления, записи, открытия и закрытия файлов также имеется обширная категория системных вызовов (create, delete, open, close, read, write).
- Пользователям хорошо знакомы такие понятия, связанные с организацией файловой системы, как каталог, текущий каталог, корневой каталог, путь, для манипулирования которыми в операционной системе имеются системные вызовы.

Архитектурные особенности ОС.

Монолитное ядро

- По сути дела, операционная система это обычная программа, поэтому было бы логичным и организовать его так же, как устроено большинство программ, то есть составить из процедур и функций. В этом случае компоненты операционной системы являются не самостоятельными модулями, а составными частями одной большой программы. Такая структура операционной системы называется монолитным ядром (monolithic kernel).
- Монолитное ядро представляет собой набор процедур, каждая из которых может вызвать каждую. Все процедуры работают в привилегированном режиме. Таким образом, монолитное ядро это такая схема операционной системы, при которой все ее компоненты являются составными частями одной программы, используют общие структуры данных и взаимодействуют друг с другом путем непосредственного вызова процедур. Для монолитной операционной системы ядро совпадает со всей системой.

Архитектурные особенности ОС.

Монолитное ядро

- Во многих операционных системах с монолитным ядром сборка ядра, то есть его компиляция, осуществляется отдельно для каждого компьютера, на который устанавливается операционная система. При этом можно выбрать список оборудования и программных протоколов, поддержка которых будет включена в ядро. Так как ядро является единой программой, перекомпиляция это единственный способ добавить в него новые компоненты или исключить неиспользуемые. Следует отметить, что присутствие в ядре лишних компонентов крайне нежелательно, так как ядро всегда полностью располагается в оперативной памяти. Кроме того, исключение ненужных компонент повышает надежность операционной системы в целом.
- Монолитное ядро старейший способ организации операционных систем.

Архитектурные особенности ОС.

Монолитное ядро

- Даже в монолитных системах можно выделить некоторую структуру. Как в бетонной глыбе можно различить вкрапления щебенки, так и в монолитном ядре выделяются вкрапления сервисных процедур, соответствующих системным вызовам.
- Сервисные процедуры выполняются в привилегированном режиме, тогда как пользовательские программы в непривилегированном режиме. Для перехода с одного уровня привилегий на другой иногда может использоваться главная сервисная программа, определяющая, какой именно системный вызов был сделан, корректность входных данных для этого вызова, и передающая управление соответствующей сервисной процедуре с переходом в привилегированный режим работы.
- Иногда выделяют также набор программных утилит, которые помогают выполнению сервисных процедур.

Архитектурные особенности ОС.

Слоеные системы (Layered systems)

- Продолжая структуризацию, можно разбить всю вычислительную систему на ряд более мелких уровней с хорошо определенными связями между ними, так чтобы объекты уровня N могли вызывать только объекты из уровня $N-1$. Нижним уровнем в таких системах обычно является hardware, верхним уровнем интерфейс пользователя. 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) Дейкстрой и его студентами в 1968 г.

Архитектурные особенности ОС. Слоеные системы (Layered systems)

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

Структура слоеной системы THE.

Архитектурные особенности ОС.

Слоеные системы (Layered systems)

- Слоеные системы хорошо реализуются. При использовании операций нижнего слоя не нужно знать, как они реализованы, нужно знать лишь, что они делают.
- Слоеные системы хорошо тестируются. Отладка начинается с нижнего слоя и проводится послойно. При возникновении ошибки мы можем быть уверены, что она находится в тестируемом слое.
- Слоеные системы хорошо модифицируются. При необходимости можно заменить лишь один слой, не трогая остальные. Но слоеные системы сложны для разработки: тяжело правильно определить порядок слоев, и что, к какому слою относится.
- Слоеные системы менее эффективны, чем монолитные. Так, например, для выполнения операций ввода-вывода программе пользователя придется последовательно проходить все слои - от верхнего до нижнего.

Архитектурные особенности ОС.

Виртуальные машины

- Операционная система как на виртуальной машине - это когда пользователю нет необходимости знать детали внутреннего устройства компьютера. Он работает с файлами, а не с магнитными головками и двигателем; он работает с огромной виртуальной, а не ограниченной реальной оперативной памятью; его мало волнует, единственный он на машине пользователь или нет.
- Рассмотрим несколько другой подход. Пусть операционная система реализует виртуальную машину для каждого пользователя, но, не упрощая ему жизнь, а, наоборот, усложняя. Каждая такая виртуальная машина предстает перед пользователем как абсолютно голое железо копия всего hardware в вычислительной системе, включая процессор, привилегированные и непривилегированные команды, устройства ввода-вывода, прерывания и т.д. И он один на один с этим железом. При попытке обратиться к этому виртуальному железу на уровне привилегированных команд, в действительности происходит системный вызов реальной операционной системы, которая и производит все необходимые действия. Такой подход позволяет каждому пользователю загрузить свою собственную операционную систему на виртуальную машину и делать с ней все, что душа пожелает.

Архитектурные особенности ОС. Виртуальные машины

Программа пользователя	Программа пользователя	Программа пользователя
MS-DOS	Linux	Windows-NT
Виртуальное hardware	Виртуальное hardware	Виртуальное hardware
Реальная операционная система		
Реальное hardware		

Вариант виртуальной машины.

Первой реальной системой такого рода была система CP/CMS или VM/370, как ее называют сейчас, для семейства машин IBM/370. Недостатком таких операционных систем является снижение эффективности виртуальных машин по сравнению с реальной машиной, и, как правило, они очень громоздки. Преимуществом - использование на одной вычислительной системе программ, написанных для разных операционных систем.

Архитектурные особенности ОС.

Микроядерная архитектура

- Современная тенденция в разработке операционных систем это перенесение значительной части системного кода на уровень пользователя и одновременной минимизации ядра. Речь идет о подходе к построению ядра, называемом микроядерной архитектурой (microkernel architecture) операционной системы, когда большинство ее составляющих являются самостоятельными программами.
- В этом случае взаимодействие между ними обеспечивает специальный модуль ядра, называемый микроядром. Микроядро работает в привилегированном режиме и обеспечивает взаимодействие между программами, планирование использования процессора, первичную обработку прерываний, операции ввода-вывода и базовое управление памятью.
- Остальные компоненты системы взаимодействуют друг с другом путем передачи сообщений через микроядро.

Архитектурные особенности ОС.

Микроядерная архитектура

- Основное достоинство микроядерной архитектуры высокая степень модульности ядра операционной системы. Это существенно упрощает добавление в него новых компонент. В микроядерной операционной системе можно, не прерывая ее работы, загружать и выгружать новые драйверы, файловые системы и т. д.
- Существенно упрощается процесс отладки компонент ядра, так как новая версия драйвера может загружаться без перезапуска всей операционной системы. Компоненты ядра операционной системы ничем принципиально не отличаются от пользовательских программ, поэтому для их отладки можно применять обычные средства. В то же время, микроядерная архитектура операционной системы вносит дополнительные накладные расходы, связанные с передачей сообщений, что существенно влияет на производительность.
- Для того чтобы микроядерная операционная система по скорости не уступала операционным системам на базе монолитного ядра, требуется очень аккуратно проектировать разбиение системы на компоненты, стараясь минимизировать взаимодействие между ними. Таким образом, основная сложность при создании микроядерных операционных систем необходимость очень аккуратного проектирования.

Архитектурные особенности ОС.

Смешанные системы

- Все рассмотренные подходы к построению операционных систем имеют свои преимущества и недостатки. В большинстве случаев современные операционные системы используют различные комбинации этих подходов.
- Примером смешанного подхода может служить возможность запуска операционной системы с монолитным ядром под управлением микроядра. Так устроены 4.4BSD и MkLinux, основанные на микроядре Mach. Микроядро обеспечивает управление виртуальной памятью и работу низкоуровневых драйверов. Все остальные функции, в том числе взаимодействие с прикладными программами, осуществляется монолитным ядром. Данный подход возник в результате попыток использовать преимущества микроядерной архитектуры, сохраняя по возможности хорошо отлаженный код монолитного ядра.

Архитектурные особенности ОС.

Смешанные системы

- Наиболее тесно элементы микроядерной архитектуры и элементы монолитного ядра переплетены в ядре Windows NT. Хотя Windows NT часто называют микроядерной операционной системой, это не совсем так. Микроядро NT слишком сложно и велико (более 1 Мб), чтобы носить приставку микро. Компоненты ядра Windows NT располагаются в вытесняемой памяти и взаимодействуют друг с другом путем передачи сообщений, как и положено в микроядерных операционных системах. В тоже время все компоненты ядра работают в одном адресном пространстве и активно используют общие структуры данных, что свойственно операционным системам с монолитным ядром. Кроме того, в Windows NT существует разделение между режимом ядра и режимом пользователя еще одна черта монолитного ядра. Причина всего этого проста. По мнению Microsoft, причина проста: чисто микроядерный дизайн коммерчески непрактичен, так как слишком неэффективен.
- Таким образом, Windows NT можно с полным правом назвать гибридной операционной системой.

Классификация ОС

Существует несколько схем классификации операционных систем. Ниже приведена классификация по некоторым признакам с точки зрения пользователя:

- *Реализация многозадачности*
- *Поддержка многопользовательского режима.*
- *Многопроцессорная обработка*
- *Системы реального времени*

Реализация многозадачности

По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

- многозадачные (Unix, OS/2, Windows).
- однозадачные (например, MS-DOS) и

Многозадачная ОС, решая проблемы распределения ресурсов и конкуренции, полностью реализует мультипрограммный режим([см.слайд 14](#))

Приблизительность классификации очевидна из приведенных примеров. Так в ОС MS-DOS можно организовать запуск дочерней задачи и одновременное сосуществование в памяти двух и более задач. Однако эта ОС традиционно считается однозадачной, главным образом из-за отсутствия защитных механизмов и коммуникационных возможностей

Поддержка многопользовательского режима

По числу одновременно работающих пользователей ОС можно разделить на:

- однопользовательские (MS-DOS, Windows 3.x);
- многопользовательские (Windows NT, Unix).

Наиболее существенно отличие заключается в наличии у многопользовательских систем механизмов защиты персональных данных каждого пользователя.

Многопроцессорная обработка

- Многопроцессорные системы состоят из двух или более центральных процессоров, осуществляющих параллельное выполнение команд. Поддержка мультипроцессирования является важным свойством ОС и приводит к усложнению всех алгоритмов управления ресурсами. Многопроцессорная обработка реализована в таких ОС, как Linux, Solaris, Windows NT и в ряде других.
- Многопроцессорные ОС разделяют на симметричные и асимметричные. В симметричных ОС на каждом процессоре функционирует одно и то же ядро и задача может быть выполнена на любом процессоре, то есть обработка полностью децентрализована. В асимметричных ОС процессоры неравноправны. Обычно существует главный процессор (master) и подчиненные (slave), загрузку и характер работы которых определяет главный процессор.

Системы реального времени

В разряд многозадачных ОС, наряду с пакетными системами и системами разделения времени, включаются также системы реального времени, не упоминавшиеся до сих пор.

Они используются для управления различными техническими объектами или технологическими процессами. Такие системы характеризуются предельно допустимым временем реакции на внешнее событие, в течение которого должна быть выполнена программа, управляющая объектом. Система должна обрабатывать поступающие данные быстрее, чем те могут поступать, причем от нескольких источников одновременно.

Столь жесткие ограничения сказываются на архитектуре систем реального времени, например, в них может отсутствовать виртуальная память, поддержка которой дает непредсказуемые задержки в выполнении программ.

Приведенная классификация ОС не является исчерпывающей

Процессы в ОС

Это будет далее!!!

А сегодня, спасибо за внимание!!!