

Операционные системы, среды и оболочки

Лекция 1.

Понятие операционной системы.
Основные функции ОС.

Понятие операционной системы

- **Операционная система (ОС)** – это комплекс программ, обеспечивающих возможность рационального использования оборудования и другого программного обеспечения удобным для пользователя образом.
- Операционные системы призваны упростить управление ресурсами компьютера, разработку прикладного программного обеспечения и работу конечных пользователей.

Структура вычислительной системы

- В понятие вычислительной системы включают:
 - hardware, или техническое обеспечение: процессор, память, монитор, дисковые устройства и т.д., объединенные магистральным соединением, которое называется шиной.
 - software, или программное обеспечение: системное, прикладное, средства разработки и т.д. К прикладному программному обеспечению, как правило, относятся разнообразные банковские и прочие бизнес-программы, игры, текстовые процессоры и т. п. Под системным программным обеспечением обычно понимают программы, способствующие функционированию и разработке прикладных программ. Деление на прикладное и системное программное обеспечение является отчасти условным и зависит от того, кто осуществляет такое деление.

Техническое обеспечение вычислительных систем

- **Основная память** используется для запоминания программ и данных в двоичном виде и организована в виде упорядоченного массива ячеек, каждая из которых имеет уникальный цифровой адрес. Типовые операции над основной памятью – считывание и запись содержимого ячейки с определенным адресом.
- Выполнение различных операций с данными осуществляется изолированной частью компьютера, называемой **центральным процессором** (ЦП). ЦП также имеет ячейки для запоминания информации, называемые регистрами. Их разделяют на регистры общего назначения и специализированные регистры. В современных компьютерах емкость регистра обычно составляет 4–8 байт. Регистры общего назначения используются для временного хранения данных и результатов операций. Для обработки информации обычно организовывается передача данных из ячеек памяти в регистры общего назначения, выполнение операции центральным процессором и передача результатов операции в основную память.
- Специализированные регистры используются для контроля работы процессора. Наиболее важными являются: программный счетчик, регистр команд и регистр, содержащий информацию о состоянии программы.

ЦП

Память

Носители
информации

Принтер

Монитор

Клавиатура

Системная магистраль

Взаимодействие с периферийными устройствами

- Периферийные устройства предназначены для ввода и вывода информации. Каждое устройство обычно имеет в своем составе специализированный компьютер, называемый **контроллером** или адаптером. Когда контроллер вставляется в разъем на материнской плате, он подключается к шине и получает уникальный номер (адрес). После этого контроллер осуществляет наблюдение за сигналами, идущими по шине, и отвечает на сигналы, адресованные ему.
- Любая операция ввода-вывода предполагает диалог между ЦП и контроллером устройства. Когда *процессору* встречается команда, связанная с вводом-выводом, входящая в состав какой-либо программы, он выполняет ее, посылая сигналы контроллеру устройства. Это так называемый программируемый ввод-вывод.

Взаимодействие с периферийными устройствами

- В свою очередь, любые изменения с внешними устройствами имеют следствием передачу сигнала от устройства к ЦП. С точки зрения ЦП это является асинхронным событием и требует его реакции.
- Для того чтобы обнаружить такое событие, между машинными циклами *процессор* опрашивает специальный регистр, содержащий информацию о типе устройства, сгенерировавшего сигнал. Если сигнал имеет место, то ЦП выполняет специфичную для данного устройства программу, задача которой – отреагировать на это событие надлежащим образом (например, занести символ, введенный с клавиатуры, в специальный буфер).
- Такая программа называется программой обработки *прерывания*, а само событие *прерыванием*, поскольку оно нарушает плановую работу *процессора*. После завершения обработки *прерывания* *процессор* возвращается к выполнению программы. Эти действия компьютера называются вводом-выводом с использованием *прерываний*.

Программное обеспечение ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

- Программы хранятся в виде последовательности машинных команд, которые должен выполнять центральный *процессор*. Каждая команда состоит из поля операции и полей операндов, то есть тех данных, над которыми выполняется данная операция. Весь набор машинных команд называется машинным языком.
- Выполнение программы осуществляется следующим образом:
 - Машинная команда, на которую указывает программный счетчик, считывается из памяти и копируется в регистр команд.
 - Она декодируется, после чего исполняется.
 - После выполнения команды программный счетчик указывает на следующую команду.
- Эти действия, называемые *машинным циклом*, затем повторяются.

Основные функциональные задачи ОС

- Операционные системы, как часть системного программного обеспечения, выполняет ряд важных задач:
 - организация программного интерфейса;
 - организация программно-аппаратного взаимодействия (взаимодействие с аппаратурой);
 - организация пользовательского интерфейса;
 - организация межмашинного взаимодействия.

Операционная система как виртуальная машина

- При разработке ОС широко применяется абстрагирование, которое является важным методом упрощения и позволяет сконцентрироваться на взаимодействии высокоуровневых компонентов системы, игнорируя детали их реализации. В этом смысле ОС представляет собой интерфейс между пользователем и компьютером.
- Архитектура большинства компьютеров на уровне машинных команд очень неудобна для использования прикладными программами.
- Например, работа с диском предполагает знание внутреннего устройства его электронного компонента – контроллера для ввода команд вращения диска, поиска и форматирования дорожек, чтения и записи секторов и т. д.
- Современные вычислительные комплексы можно создать иллюзию неограниченного размера операционной памяти и числа процессоров.
- 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 сложной архитектуры компьютера.
- Представим, к примеру, что произойдет, если несколько программ, работающих на одном компьютере, будут пытаться одновременно осуществлять вывод на принтер.
- Операционная система предотвращает такого рода хаос 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.
- Операционная система, как менеджер ресурсов, осуществляет упорядоченное и контролируемое распределение процессоров, памяти и других ресурсов между различными программами.

Операционная система как постоянно функционирующее ядро

- Операционная система – это программа, постоянно работающая на компьютере и взаимодействующая со всеми прикладными программами (резидентные программы).
- 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 гг.). Ламповые машины. Операционных систем нет
 - В середине 40-х были созданы первые ламповые вычислительные устройства и появился принцип программы, хранящейся в памяти машины (John Von Neumann, июнь 1945 г.). Программирование осуществлялось исключительно на машинном языке. Об операционных системах не было и речи, все задачи организации вычислительного процесса решались вручную каждым программистом с пульта управления. Программа загружалась в память машины в лучшем случае с колоды перфокарт, а обычно с помощью панели переключателей.
 - Вычислительная система выполняла одновременно только одну операцию (ввод-вывод или собственно вычисления). Отладка программ велась с пульта управления с помощью изучения состояния памяти и регистров машины. В конце этого периода появляется первое системное программное обеспечение: в 1951–1952 гг. возникают прообразы первых компиляторов с символических языков (Fortran и др.), а в 1954 г. Nat Rochester разрабатывает Ассемблер для IBM-701.
 - Существенная часть времени уходила на подготовку запуска программы, а сами программы выполнялись строго последовательно. Такой режим работы называется последовательной обработкой данных. В целом первый период характеризуется крайне высокой стоимостью вычислительных систем, их малым количеством и низкой эффективностью использования.

Краткая история эволюции ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

- Второй период (1955 г.–начало 60-х). Компьютеры на основе транзисторов. Пакетные операционные системы
 - С середины 50-х годов начался следующий период в эволюции вычислительной техники, связанный с появлением новой технической базы – полупроводниковых элементов. Применение транзисторов вместо часто перегоравших электронных ламп привело к повышению надежности компьютеров. Одновременно наблюдается бурное развитие алгоритмических языков (LISP, COBOL, ALGOL-60, PL-1 и т.д.). Появляются первые настоящие компиляторы, редакторы связей, библиотеки математических и служебных подпрограмм. Упрощается процесс программирования.
 - Изменяется сам процесс прогона программ. Теперь пользователь приносит программу с входными данными в виде колоды перфокарт и указывает необходимые ресурсы. Такая колода получает название задания. Оператор загружает задание в память машины и запускает его на исполнение. Полученные выходные данные печатаются на принтере, и пользователь получает их обратно через некоторое (довольно продолжительное) время.
 - Смена запрошенных ресурсов вызывает приостановку выполнения программ, в результате процессор часто простаивает. Для повышения эффективности использования компьютера задания с похожими ресурсами начинают собирать вместе, создавая пакет заданий.
 - Появляются первые системы пакетной обработки, которые просто автоматизируют запуск одной программы из пакета за другой и тем самым увеличивают коэффициент загрузки процессора. При реализации систем пакетной обработки был разработан формализованный язык управления заданиями, с помощью которого программист сообщал системе и оператору, какую работу он хочет выполнить на вычислительной машине.

Краткая история эволюции вычислительных систем

- Третий период (начало 60-х – 1980 г.). Компьютеры на основе интегральных микросхем. Первые многозадачные ОС
 - Следующий важный период развития вычислительных машин относится к началу 60-х – 1980 г. В это время в технической базе произошел переход от отдельных полупроводниковых элементов типа транзисторов к интегральным микросхемам. Растет сложность и количество задач, решаемых компьютерами. Повышается производительность процессоров.
 - Повышению эффективности использования процессорного времени мешает низкая скорость работы механических устройств ввода-вывода (быстрый считыватель перфокарт мог обработать 1200 перфокарт в минуту, принтеры печатали до 600 строк в минуту). Вместо непосредственного чтения пакета заданий с перфокарт в память начинают использовать его предварительную запись, сначала на магнитную ленту, а затем и на диск.
 - Когда в процессе выполнения задания требуется ввод данных, они читаются с диска. Точно так же выходная информация сначала копируется в системный буфер и записывается на ленту или диск, а печатается только после завершения задания. Вначале действительные операции ввода-вывода осуществлялись в режиме off-line, то есть с использованием других, более простых, отдельно стоящих компьютеров.
 - Магнитные ленты были устройствами последовательного доступа, то есть информация считывалась с них в том порядке, в каком была записана. Появление магнитного диска, для которого не важен порядок чтения информации, то есть устройства прямого доступа, привело к дальнейшему развитию вычислительных систем. При обработке пакета заданий на магнитной ленте очередность запуска заданий определялась порядком их ввода. При обработке пакета заданий на магнитном диске появилась возможность выбора очередного выполняемого задания. Пакетные системы начинают заниматься планированием заданий: в зависимости от наличия запрошенных ресурсов, срочности вычислений и т.д. на счет выбирается то или иное задание.

Краткая история эволюции ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

- Третий период (начало 60-х – 1980 г.). Компьютеры на основе интегральных микросхем. Первые многозадачные ОС
 - Идея **мультипрограммирования** заключается в следующем: пока одна программа выполняет операцию ввода-вывода, процессор не простаивает, как это происходило при однопрограммном режиме, а выполняет другую программу. Когда операция ввода-вывода заканчивается, процессор возвращается к выполнению первой программы.
 - Мультипрограммирование при реализации ОС позволяет решить следующие задачи:
 - Организация интерфейса между прикладной программой и ОС при помощи системных вызовов.
 - Организация очереди из заданий в памяти и выделение процессора одному из заданий потребовало планирования использования процессора.
 - Переключение с одного задания на другое требует сохранения содержимого регистров и структур данных, необходимых для выполнения задания, иначе говоря, контекста для обеспечения правильного продолжения вычислений.
 - Поскольку память является ограниченным ресурсом, нужны стратегии управления памятью, то есть требуется упорядочить процессы размещения, замещения и выборки информации из памяти.
 - Организация хранения информации на внешних носителях в виде файлов и обеспечение доступа к конкретному файлу только определенным категориям пользователей.
 - Поскольку программам может потребоваться произвести санкционированный обмен данными, необходимо их обеспечить средствами коммуникации.
 - Для корректного обмена данными необходимо разрешать конфликтные ситуации, возникающие при работе с различными ресурсами и предусмотреть координацию программами своих действий, т.е. снабдить систему средствами синхронизации.

Краткая эволюция ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

- Четвертый период (с 1980 г. по настоящее время). Персональные компьютеры. Классические, сетевые и распределенные системы
 - Следующий период в эволюции вычислительных систем связан с появлением больших интегральных схем (БИС). Компьютер, не отличающийся по архитектуре от PDP-11, по цене и простоте эксплуатации стал доступен отдельному человеку, а не отделу предприятия или университета. Наступила эра персональных компьютеров. Первоначально персональные компьютеры предназначались для использования одним пользователем в однопрограммном режиме, что повлекло за собой деградацию архитектуры этих ЭВМ и их операционных систем (в частности, пропала необходимость защиты файлов и памяти, планирования заданий и т. п.).
 - В *сетевых операционных системах* пользователи могут получить доступ к ресурсам другого сетевого компьютера, только они должны знать об их наличии и уметь это сделать. Каждая машина в сети работает под управлением своей локальной операционной системы, отличающейся от операционной системы автономного компьютера наличием дополнительных средств (программной поддержкой для сетевых интерфейсных устройств и доступа к удаленным ресурсам), но эти дополнения не меняют структуру операционной системы.
 - *Распределенная система*, напротив, внешне выглядит как обычная автономная система. Пользователь не знает и не должен знать, где его файлы хранятся – на локальной или удаленной машине – и где его программы выполняются. Он может вообще не знать, подключен ли его компьютер к сети. Внутреннее строение распределенной операционной системы имеет существенные отличия от автономных систем.

Основные понятия, концепции ОС

- **Системные вызовы**

- В любой *операционной системе* поддерживается механизм, который позволяет пользовательским программам обращаться к услугам ядра ОС.
- **Системные вызовы** (system calls) – это 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 исполняется он 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.

Основные понятия ОС

- **Прерывания**

- Прерывание (hardware interrupt) – это событие, генерируемое внешним (по отношению к процессору) устройством. Посредством аппаратных прерываний аппаратура либо информирует центральный процессор о том, что произошло 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.

Основные понятия ОС

- **Исключительные ситуации**

- Исключительная ситуация (exception) – событие, возникающее в результате попытки выполнения программой команды, которая по каким-то причинам не может быть выполнена до конца.
- Примерами таких команд могут быть попытки доступа к ресурсу при отсутствии достаточных привилегий или обращения к отсутствующей странице памяти.
- Исключительные ситуации, 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 выполняться дальше. Возникновение в процессе работы операционной системы исправимых исключительных ситуаций считается нормальным явлением.
 - Неисправимые исключительные ситуации чаще всего возникают в результате ошибок в программах (например, деление на ноль). Обычно в таких случаях операционная система реагирует завершением программы, вызвавшей исключительную ситуацию.

Основные понятия ОС

- **Файлы**

- Файлы предназначены для хранения информации на внешних носителях, то есть принято, что информация, записанная, например, на диске, должна находиться внутри файла. Обычно под файлом понимают именованную часть пространства на носителе информации.
- Главная задача **файловой системы** (file system) – скрывание особенностей ввода-вывода и возможность дать программисту простую абстрактную модель файлов, независимых от устройств.
- Для чтения, создания, удаления, записи, открытия и закрытия файлов также имеется обширная категория системных вызовов (создание, удаление, открытие, закрытие, чтение и т.д.).
- Пользователям хорошо знакомы такие связанные с организацией файловой системы понятия, как каталог, текущий каталог, корневой каталог, путь. Для манипулирования этими объектами в операционной системе имеются системные вызовы.

Основные понятия ОС

- Процессы, нити
 - Понятие **процесса** характеризует некоторую совокупность набора исполняющихся команд, ассоциированных с ним ресурсов (выделенная для исполнения память или адресное пространство, стеки, используемые файлы и устройства ввода-вывода и т. д.) и текущего момента его выполнения (значения регистров, программного счетчика, состояние стека и значения переменных), находящуюся под управлением операционной системы.
 - Не существует взаимно-однозначного соответствия между *процессами* и программами, обрабатываемыми вычислительными системами. Даже в случае обработки только одной программы в рамках одного *процесса* нельзя считать, что *процесс* представляет собой просто динамическое описание кода исполняемого файла, данных и выделенных для них ресурсов.
 - *Процесс* находится под управлением операционной системы, поэтому в нем может выполняться часть кода ее ядра, как в случаях, специально запланированных авторами программы (например, при использовании системных вызовов), так и в непредусмотренных ситуациях (например, при обработке внешних прерываний).

Основные функции классической ОС

- Шесть основных функций, которые выполняют классические операционные системы:
 - 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.

Классификация ОС

- **Реализация многозадачности**
- По числу одновременно выполняемых задач операционные системы можно разделить на два класса:
 - многозадачные (Unix, OS/2, Windows);
 - однозадачные (например, MS-DOS).
- **Многозадачная ОС**, решая проблемы распределения ресурсов и конкуренции, полностью реализует мультипрограммный режим.
 - Многозадачный режим, который воплощает в себе идею деления времени, называется вытесняющим (preemptive). Каждой программе выделяется квант процессорного времени, по истечении которого управление передается другой программе. Говорят, что первая программа будет вытеснена. В вытесняющем режиме работают пользовательские программы большинства коммерческих ОС.
 - В некоторых ОС (Windows 3.11, например) пользовательская программа может монополизировать процессор, то есть работает в невытесняющем режиме. Как правило, в большинстве систем не подлежит вытеснению код собственно ОС. Ответственные программы, в частности задачи реального времени, также не вытесняются. Более подробно об этом рассказано в лекции, посвященной планированию работы процессора.
 - По приведенным примерам можно судить о приблизительности классификации. Так, в ОС MS-DOS можно организовать запуск дочерней задачи и наличие в памяти двух и более задач одновременно. Однако эта ОС традиционно считается однозадачной, главным образом из-за отсутствия защитных механизмов и коммуникационных возможностей.

Классификация ОС

- **Поддержка многопользовательского режима**
- По числу одновременно работающих пользователей ОС можно разделить на:
 - однопользовательские (MS-DOS, Windows 3.x);
 - многопользовательские (Windows 2000, XP, Unix).
- Наиболее существенное отличие между этими ОС заключается в наличии у многопользовательских систем механизмов защиты персональных данных каждого пользователя.

Классификация ОС

- **Многопроцессорная обработка**
- Вплоть до недавнего времени вычислительные системы имели один центральный процессор. В результате требований к повышению производительности появились многопроцессорные системы, состоящие из двух и более процессоров общего назначения, осуществляющих параллельное выполнение команд.
- Поддержка мультипроцессорирования является важным свойством ОС и приводит к усложнению всех алгоритмов управления ресурсами. Многопроцессорная обработка реализована в таких ОС, как Linux, Solaris, Windows NT, и 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. При этом каждому из процессоров доступна вся память.
 - В *асимметричных* ОС процессоры неравноправны. Обычно существует главный процессор (master) и подчиненные (slave), загрузку и характер работы которых определяет главный процессор.

Классификация ОС

- **Системы реального времени**
- В разряд многозадачных ОС, наряду с пакетными системами и системами разделения времени, включаются также **системы реального времени**.
 - 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 они 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.