**1) Программное обеспечение ЭВМ. Его типы.**

**Программное обеспечение** - совокупность

программных средств, для обеспечения нормальной работы вычислительной системы.

Программное обеспечение ЭВМ включает в себя:

1)операционные системы, в составе которых есть:

а) управляющие программы (супервизоры);

б) системные обрабатывающие программы (трансляторы);

2)сервисные программы, или программы технического обслуживания;

3)инструментальный язык, или язык операционной системы;

4)системы (языки) программирования: машинно-ориентированные, программно-ориентированные и процедурно-ориентированные пакеты прикладных программ.

Основной частью любого программного обеспечения является операционная система.

**Типы операционных систем**

**1.Системное ПО:**

**1)**Базовое ПО(Операционные системы, Оболочки ОС, Сетевые ОС)

2)Сервисное ПО(Поисковые в сети, Диагностика,Антивирусные программы,Архиваторы)

**2.Инструментальное ПО**

1)Языки и системы программирования

2)Интегрированные среды программирования

3)Программные комплексы

**3.Прикладное ПО**

1)Общего назначения

2)Мультимедиа

3)Проблемно-ориентированные

4)информационно поисковые

5)издательские системы

6)прочие

**2. Понятие операционной системы. Функции и задачи ОС. Основные части ОС, взаимодействие с пользователем.**

***Операционная система (ОС)***- это комплекс программ, которые загружаются при включении компьютера. ОС является неотъемлемой частью ЭВМ и является посредником между компьютером и пользователем.

**Функции операционной системы**   
1)Производит диалог с пользователем.   
2)Осуществляет управление компьютером и его ресурсами (оперативной памятью, местом на дисках).   
3)Запускает прикладные и пользовательские программы на выполнение.   
ОС обеспечивает пользователю и прикладным программам удобный способ общения с устройствами компьютера.

***Любая операционная система включает в себя три основных элемента, взаимодействующих с пользователем:***

1)Файловая система.

2)Драйверы внешних устройств.

3)Процессор командного языка.

В **составе операционной системы** принято выделять следующие части:

**1)базовый модуль** (ядро) - управляет файловой системой, обеспечивает доступ к ней и обмен файлами между периферийными устройствами. К основным функциям ядра операционной системы относятся:

**1].**инициализация системы (загрузка ядра в оперативную память и его запуск);

**2].**управление процессами (создание, завершение и отслеживание существующих процессов);

**3].**управление памятью (отображение виртуальной памяти процессов в физическую оперативную память компьютера, которая имеет ограниченные размеры);

**4].**управление файлами (создание модели файловой системы – иерархии каталогов и файлов);

**5].**коммуникационные средства (обмен данными между процессами, выполняемыми внутри одного компьютера, в различных узлах локальной или глобальной сети передачи данных;

**6].**программный интерфейс (доступ к возможностям ядра со стороны пользовательских процессов).

**2)командный процессор** - расшифровывает и исполняет любые действия или команды пользователя, поступающие в систему;

**3)драйверы периферийных устройств** - обеспечивают согласованность работы периферийных устройств с ОС. **Драйвер** – это программа, обеспечивающая взаимодействие ОС с устройством ПК или периферийным устройством. В функции драйвера входит обработка прерываний устройства, управление очередью запросов к нему, преобразование запросов в команды управления устройством;

**4)дополнительные сервисные программы (утилиты) -** служат для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики и тестирования аппаратных и программных средств, оптимизации использования дискового пространства, восстановления разрушенной на магнитном диске информации и т.п.).

**8. Кэш память**

**Кэш память** - представляет собой буферную память микроЭВМ. Между 2 устройствами с разными скоростями и выполняет функцию выравнивания скоростей передачи данных.

Есть 5 кэш буферов.

1) Процессорная память

2) Кэш(L1.2.3)

3) Программный кэш- часть ОП

4) Аппаратный кэш

5) Внутренний кэш устройства

**3. Типы и классы операционных систем.**

В различных моделях ПЭВМ используются ОС с разной

архитектурой и возможностями; для их работы необходимы различные ресурсы оперативной памяти; они предоставляют разную степень сервиса для пользователя.

**1.Самые первые** ОС для ПК разрабатывались для 8-ми разрядных микропроцессоров; они предоставляли пользователю лишь самый необходимый набор средств управления ПК. Обеспечение сервиса, «дружественного интерфейса», поддержка внешних устройств возлагалась на прикладные программы. ОС этого класса не давали никаких особых возможностей для системного программиста. *ПРИМЕР:* ОС *СР/М* (1974 г.) - это первая ОС для микроЭВМ.

**2.Второй класс** - ОС с более развитыми средствами доступа к аппаратным компонентам, гибкой файловой системой, основанной на иерархии каталогов, удобным для пользователя командным языком. Эти операционные системы разрабатывались для 16-ти разрядных (а потом и 32-х разрядных) микропроцессоров. Средства, предоставляемые ОС этого класса, позволяют формировать удобный сервис для разработки программного обеспечения и создания рабочих мест. *ПРИМЕР:* ОС *MS DOS* и семейство *DOS*.

**3.Третий класс** ОС ориентирован на эффективную поддержку разработки программного обеспечения и одновременное выполнение нескольких задач. У таких ОС имеется развитая файловая система, мощный командный язык, обеспечивается программирование доступа ко всем типам внешних устройств. В состав этих ОС входит множество служебных программ - утилит, обеспечивающих выполнение разнообразных функций. В этих ОС заложена возможность одновременной работы с ЭВМ нескольких пользователей с отдельных терминалов и обработки сразу нескольких задач. Системы этого класса требуют значительных ресурсов памяти и быстродействия. *ПРИМЕР:*ОС *UNIX, OS/2.*

**4.К четвертому классу** относятся ОС, ориентированные на поддержку удобной работы конечных пользователей. Эти системы имеют развитые средства поддержки диалога - так называемый *графический интерфейс:*дисплейные окна, разнообразные графические объекты, активно используют манипуляторы курсора для выбора объектов и операций над ними. Имеются средства для поддержания многозадачного режима работы.

*ПРИМЕР:* *Windows’95, 98, 2000. Windows NT, XP.*

Все операционные системы можно разделить на две группы:

***1.)многопользовательские*** - предусматривающие одновременную работу и обращение к системным ресурсам нескольких пользователей с отдельных терминалов, подключенных к общей ЭВМ или серверу. Примером такой ОС может быть система UNIX.

***2.)однопользовательские*** - предусматривающие работу за компьютером или на рабочей станции только одного пользователя.

В свою очередь, они подразделяются на три группы:

***1)однозадачные*** - т.е. способные выполнять в любой момент времени только одну пользовательскую, системную или прикладную задачу. Классическим примером таких ОС является система MS DOS и DOS - подобные системы.

***2)сетевые*** - работающие с несколькими ПЭВМ, объединенными в компьютерную сеть: Novell NetWare, Windows NT и др.

***3)многозадачные*** - позволяющие на одном компьютере в любой момент времени выполнять одновременно несколько задач, например, распечатывать текст на принтере, редактировать документ и выполнять вычисления. К таким системам относятся Windows’95 и OS/2.

*По реализации многозадачности все многозадачные ОС делятся на две группы:*

***1)с кооперативной многозадачностью*** - это ОС, где задачи сами следят за временем своего выполнения, а потом передают управление другой задаче (так называемое *переключение по событию*). То есть, если задача сама не отдает ресурсы, то у неё их «отнять» нельзя. Такими системами являются DOS+ Windows 3.11, Windows’95, Novell NetWare.

***2) c вытесняющей многозадачностью*** - в таких ОС системные ресурсы выделяются задаче специальным диспетчером в зависимости от приоритетов задачи и ресурса на некоторое непродолжительное время (квант времени ~ 0.1 мксек.), а потом эта задача принудительно выгружается. Таким образом, у пользователя и прикладной программы создается полная иллюзия одновременного выполнения нескольких задач. Такие ОС называются истинно многозадачными. Это OS/2 Warp, Windows NT, UNIX.

**5. Понятие архитектуры микроЭВМ. BIOS. Этапы загрузки ОС.**

**Понятие архитектуры микроЭВМ**

Базовая система ввода-вывода (BIOS)   
Драйверы стандартных устройств образуют в совокупности базовую систему ввода - вывода - BIOS -сокращение от Basic Input/Output System, которая занесена в ПЗУ (ROM)- постоянное запоминающее устройство системного блока ПК.   
ROM-BIOS является самым низким уровнем, находящемся ниже всех других программ и операций в компьютере. Ее задача – заботиться о сиюминутных потребностях аппаратуры компьютера и избавить все другие программы от знания деталей того, как это делается. BIOS является программой, хотя и довольно сложной. Функционально она является своеобразным мостом между аппаратурой компьютера и нашими программами.

Особенность системы BIOS состоит в том , что она работает непосредственно с аппаратурой компьютера, используя много практических знаний о том, как эта аппаратура функционирует.

Система BIOS разделена на три функциональные части.

***Первая функциональная часть*** – начальные программы, которые запускают наш компьютер, когда мы включаем электропитание. В свою очередь стартовая программа может быть разделена на две основные части.

**Первая** – подпрограмма самотестирования аппаратуры, которая проверяет работоспособность компонентов компьютера. Тестируется оперативная память компьютера с целью обнаружения дефектов, делается проверка внешней памяти и другие проверки. Эти действия необходимы для того, чтобы убедиться, что компьютер работает без сбоев.

**Вторая** – подпрограмма инициализации. Инициализация состоит в установке вектора прерывания. Когда случается прерывание, компьютер переключается на ту программу, которая обрабатывает прерывания.

Система BIOS содержит сведения обо всех компонентах стандартного оборудования, которое может иметь IBM PC, и она выполняет ***инициализацию***, которую требует каждая из них. При инициализации определяется конфигурация компьютера, и эти сведения передаются в ROM-BIOS.

Существует специальное оборудование, которое требует соей собственной специальной программной поддержки. Это становится возможным за счет механизма, который позволяет основной системе BIOS найти и распознать *программу расширения.*

**Последнюю часть** системы BIOS составляет программа boot record (начальная загрузка), которая пытается загрузить DOS или любую другую операционную систему, которая может быть использована в нашем компьютере. Процесс начальной загрузки начинается с попытки системы BIOS считать программу загрузки с начала магнитного диска.

Система BIOS сначала пытается обнаружить эту программу на жёстком диске (обычно диск С), если это не удаётся, то эта попытка повторяется, но уже с другим диском. Если ни с одного диска нельзя прочитать программу загрузки, то BIOS входит в свой «недисковый режим»: это означает либо активизацию встроенного Бейсика либо к печати сообщения, говорящего, что компьютер нуждается в диске с программой начальной загрузки.

**Загрузчик операционной системы**

***Загрузчик*** – это очень короткая программа, находящаяся в первом секторе каждого диска с операционной системой. Функция этой программы заключается в считывании в память ядра операционной системы.

На жестком диске загрузчик ОС состоит из двух частей. Это связано с тем, что жесткий диск может быть разбит на несколько разделов (логических дисков). Первая часть загрузчика находится в первом секторе жесткого диска, она выбирает , с какого из разделов жесткого диска следует продолжить загрузку. Вторая часть загрузчика находится в первом секторе этого раздела, она считывает ядро операционной системы и передает ему управление.

**Загрузка ОС.**

При включении электропитания компьютера начинают работать программы проверки оборудования, находящиеся в постоянной памяти компьютера. Если они находят ошибку, то выводят код ошибки на экран .

После окончания тестирования программа начальной загрузки пытается прочесть с дискеты, установленной на дисководе, программу – загрузчик ОС ( ВООТ ) ( расположенная в начале диска ).После того, как с диска , с которого загружалась ОС, прочитана программа – загрузчик ОС, эта программа считывает в память модули ОС ( базовый модуль и ему передается управление).

В состав базового модуля входит основной загрузчик, компьютер ищет остальные модули ОС и считывает их в ОЗУ. После окончания загрузки ОС управление передается командному процессору, на экране появляется приглашение системы к вводу команд пользователя.

**25.Обработка страничных прерываний. Сегментная и симентно-страничная организация памяти.**

Когда процесс обращается по некоторому адресу, а содержащая его страница отсутствует в физической памяти, то система ***генерирует страничное прерывание*** (прерывание по особому случаю в странице).

В случае сигм-стр. распределения памяти весь образ процесса делится на сегменты при этом с каждым сегментом связываются разные права доступа.

После размещения по сегментам , они делятся на страницы по этому недостающая информация заполняется нулями в последней странице сегмента.

У каждого процесса есть таблица сегмента и таблица страниц.

**17.Диспетчирезация процессов. Основные алгоритмы диспетчеризации.**

**Диспетчеризация процессов**- представляет собой процедуру выделения времени процессора, процессу на активность.

В основе алгоритма планирования процессов лежит концепция квантования времени. В соответствии с этой концепцией каждому потоку поочерёдно предоставляется квант времени.

Смена активности происходит при 1 из 4 условий.

1) Если поток завершился и покинул систему.

2) Если произошла ошибка.

3) Если поток перешёл в сост. ошибки.

4)Если истек квант процессорного времени.

Поток который исчерпал свой квант времени переходит в состояние готовности.

**18.Алгоритмы сервисных запросов WAIT и SIGNAL.**

WAIT и SIGNAL представляют собой 2 основных серверных запроса. Первая программа- прерывание. Сигнал WAIT указывает ЦП, что адресованная ячейка памяти или устройство ввода вывода ещё не готово к передаче данных. ЦП генерирует состояние ожидания (холостые такты, в которых не происходит никаких изменений с ЦП) до тех пор, пока активен этот сигнал.

Любое обращение к ядру сопровождается запросом Wait.

Wait (SVC 0) ; SIGNAL (SVC 1)

**9. Прерывания . Обработка Аппаратных прерываний**

**Прерывание** – это важная процедура, которая позволяет изменить нормальную последовательность команд, выполняемых процессором.

Сущность прерывания заключается в следующем:

Устройство, которое требует внимания процессора, сообщает об этом с помощью специального сигнала (запрос на прерывание). По этому сигналу управление CPU передается ОС. ОС запоминает состояние прерванного процесса и хранит эту информацию в спец. регистре микропроцессора. Такой регистр называется СТЕК. Затем ОС анализирует, от какого устройства произошло прерывание и затем передает управление программе, которая управляет устройством, выдавшим запрос на прерывание.

Такая программа называется «**обработчик прерывания**».

Прерывание может быть вызвано не только каким-нибудь устройством, но и выполняющимся процессом.

В начале прерывания использовались в основном для управления процессором устройствами ввода-вывода. Затем прерывания стали использовать для организации внутренней работы ЭВМ. В соответствии с этим существуют следующие типы прерываний:

1. **Аппаратные прерывания** – прерывания от устройств компьютера.

2. **Программные прерывания** – прерывания, которые вырабатывают процессы, находящиеся на стадии выполнения.

3.**Логические прерывания** – Эти прерывания вырабатывает сам процессор, когда встречается с каким-либо необходимым условием:

а) деление на 0

б) переполнение регистров микропроцессора

в) пошаговое выполнение программ

г) режим контрольных точек.

Каждое прерывание имеет два параметра:

1)Номер прерывания 2)Вектор прерывания.

**Вектор прерывания** – это адрес ячейки памяти, где хранится программа – обработчик прерывания. Прерывания обозначаются - **IRQ**.

**Обработка Аппаратных прерываний**

**Обработка прерывания в реальном (однозадачном) режиме** производится в три этапа:

**1) прекращение выполнения текущей программы**;

Должно произойти так, чтобы потом вернуться и продолжить работу. Для этого необходимо сохранить содержимое регистров, так как они являются ресурсами, разделяемыми между программами.

Эти регистры сохраняются микропроцессором автоматически. Наиболее удобным местом хранения регистров является стек.

**2) переход к выполнению и выполнение программы обработки прерывания;**

Здесь определяется источник прерывания и вызывается соответствующий обработчик прерывания.

В реальном режиме микропроцессора допускается 256 источников - по кол-ву элементов таблицы векторов прерываний.

**3) возврат управления прерванной программе.**

Необходимо привести стек в состояние, в котором он был сразу после передачи управления данной процедуре.

**Работа системы прерываний в защищенном (многозадачном) режиме.**

Обработка прерываний в защищенном режиме отличается от обработки в реальном режиме так же сильно, как и защищенный режим отличается от реального. Потому что:

1. В защищенном режиме немного изменено распределение номеров векторов прерываний.

2. Принципиально иным является механизм обработки прерываний.

**Классификация прерываний в защищенном режиме**

Прерывания и исключения можно разделить на несколько групп:

 сбой;

 ловушка;

 аварийное завершение.

**Сбой (ошибка)** — прерывание или исключение, при возникновении которого в стек (в память) записываются значения регистров, указывающие на команду, вызвавшую данное прерывание.

Это позволяет, получив доступ к сегменту кода, исправить ошибочную команду в обработчике прерывания и, вернув управление программе, фактически осуществить ее рестарт (в реальном режиме при возникновении прерывания в стеке всегда запоминается адрес команды, следующей за той, которая вызвала это прерывание).

**Стековой называют память**, доступ к которой организован по принципу: "последним записан - первым считан" (Last Input First Output - LIFO)

**Ловушка** — прерывание или исключение, при возникновении которого в стек записываются значения регистров, указывающие на команду, следующую за командой, вызвавшей данное прерывание.

Так же, как и в случае ошибок возможен рестарт программы..

**Аварийное завершение** — прерывание, при котором информация о месте его возникновения недоступна или неполна и поэтому рестарт практически невозможен, если только данная ситуация не была запланирована заранее.

**6.Основные аппаратные подсистемы микроЭВМ**.

1)Процессорная подсистема (ЦП, кэш L1)

2)Кэш L2. Микросхема на сист. плате, которая быстрее микросхем ОП.

3)ОП выполненая на основе DRAM

4) Шины (Локальная шина, работающая на частоте процессора; шина расширения (PSI)

5) Графическая подсистема, в состав адаптера входит: граф. контролер, граф. процессоры, видео память.

6)Дисковая поддержка

7)Набор портов или слотов, который обуславливает открытую конфигурацию компьютера.

**16.Понятие процесса и потока. Состояние процесса.**

**Понятия процесса и потока. Состояния процесса**

Под процессом понимается любая деятельность реализуемая на процессоре, приводимая к результату. Процесс представляет собой основную логическую единицу ОС, претендуемую на использование тех или иных ресурсов системы.

Процесс проходит несколько стадий:

1) порождение 2) завершение

**Интервал существования процесса**- время между зарождение и завершением.

Возможные состояния процесса:

- действие (использует процессор в данный момент)

- готовность (приостановлен, чтобы позволить выполняться другому процессу)

- блокировка (не может быть запущен прежде, чем произойдет некое внешнее событие)

**Поток**- более мелкая программная единица по сравнения с процессом.

Процесс делится в системе на несколько потоков , представляемые собой последовательность команд.

Потоку выделяется только процессорное время , а процессу всё кроме времени.

**11.Понятие виртуального ресурса и виртуальной ЭВМ. Иерархия виртуальных машин в структуре ОС. Свойства уровней абстракции.**

**Виртуальный ресурс** представляет собой некую программно-аппаратную модель конкретного физического устройства   
Обобщение понятия виртуального ресурса приводит к концепции виртуальной машины. Виртуальная машина (ВМ) является функциональным эквивалентом реальной ЭВМ, который моделируется при помощи реальных технических средств и совокупности управляющих программ

**10.Основные принципы построения ОС**

**1)*принцип модульности***. **Модуль** – функционально законченный элемент, выполняемый в соответствии с принятым межмодульным интерфейсом. Модуль выделяется по функциональному признаку. Модульная организация позволяет легко (из)заменять неправильно работающие модули в ОС. Чаще всего используются реентерабельные и привилегированные модули.

**2)*принцип функциональной избирательности***. Для организации эффективной работы ОС, необходимо выделить некоторые модули и хранить их в ОЗУ. Эти модули составляют ядро ОС. **Ядро**:

1)Модули по управлению системы прерываний;

2)Средство управления выполнения программ (загрузка, приостановка, остановка);

3)Модули по управлению процессом (распределение процессорного времени), т.е. диспетчеры программ;

4)Модули по управлению выделения памяти. В зависимости от ОС в ядро могут ещё входить другие модули;

5)Транзитные модули (загружаются в ОЗУ по мере необходимости, при нехватке ОЗУ могут быть выгружены из памяти).

***3)принцип генерируемости*** *ОС*. Подразумевает собой возможность генерации ОС в зависимости от аппаратного обеспечения. Процесс генерации обычно производится один раз, перед достаточно долгим режимом эксплуатации. Для генерации необходимо наличие нескольких компонентов:

1) Исходный код ОС;

2) Компилятор с языка программирования на котором система написана;

3) Специальная программа и входной язык для неё, который позволяет управлять процессом генерации.

4) ОС с открытым системным кодом – Linux (UNIX), есть возможность тонкой настройки ядра для конкретного процессора.

***4)принцип функциональной избыточности***. В состав ОС должно входит несколько типов ПО для выполнения одинаковых функций (поддержка разных файловых систем).

***5)принцип виртуализации***. Позволяет представить ресурсы ОС в виде определённого набора планировщиков и мониторов и использует единую схему распределения ресурсов. Наибольшее проявление – концепция виртуальной машины (воспроизводит архитектуру реальной машины, но может обладать произвольными характеристиками).

***6)принцип независимости программ от внешних устройств*.** Связь программ с конкретным внешним устройством производится не на этапе трансляции, а на этапе выполнения программы. Получается выгода: не нужна лишняя «перекомпиляция».

***7)принцип совместимости*.** Способность выполнять программы для другой ОС или даже для другой аппаратной платформы.  
***2 уровня совместимости****:*

***1.по выполняемому коду (бинарная****).* Условия совместимости:

1) На уровне команд процессора (одна и та же платформа);

2) Совместимость на уровне системных вызовов;

3) Совместимость на уровне библиотечных вызовов, если являются динамично связываемыми.

***2. по исходному коду****.* Требуется выполнение следующих условий:

1)Наличие компилятора платформы, на котором написана программа;

2)Совместимость на уровне системных вызовов;

3)Совместимость на уровне библиотечных вызовов.

***8)принцип открытой наращиваемой ОС*** (открыт исходный код). Целостность ОС сохраняется (UNIX).

***9)принцип мобильности* (переносимости).** ОС должна легко переноситься на другую аппаратную платформу. Правила создания переносимых ОС:

1)ОС должна быть написана на языке высокого уровня, для которой существует компилятор на аппаратной платформе. В основном, современные ОС пишут на Си.

2)Необходимо избегать кода, который непосредственно работает с аппаратным обеспечением.

***10)принцип обеспечения безопасности и защиты***:

1)Защита системы от пользователя;

2)Защита от несанкционированного доступа.

В 1983 г. придуманы критерии оценки надёжности ОС. Существуют 4 класса безопасности:   
**Класс D**. Относятся системы, не удовлетворяющие системам предыдущих классов (небезопасный);   
**Класс C.** Обеспечение защиты данных от ошибок пользователя. ОС должна иметь следующие средства:   
1)Средства секретного входа;   
2)Обязательно наличие избирательного контроля доступа;   
3)Средства учёта и наблюдения (аудит);   
4)Необходима инициализация памяти при её освобождении. Современные ОС относятся к этому классу.    
**Класс B.** Основаны на помеченных данных и есть наличие распределения пользователей по категориям, любой пользователь имеет рейтинг доступа к данным.   
**Класс A.** Самый высокий уровень безопасности. Необходимо, чтобы имелось формальное (математическое) доказательство безопасности ОС. Примерно 90% процессорного времени тратится на систему безопасности. В наше время используются классы B и C.

12.**Архитектура ОС. Ядро и вспомогательные модули ОС. Режимы работы ОС**.

В состав ОС входят исполняемые и объектные модули стандартных для данной ОС форматов, библиотеки разных типов, модули исходного текста программ, программные модули специального формата (например, загрузчик ОС, драйверы ввода-вывода), конфигурационные файлы, модули справочной системы и т. д.

Большинство современных ОС представляют собой хорошо структурированные модульные системы, способные к развитию, расширению и переносу на новые платформы. Какой-либо единой архитектуры ОС не существует, но существуют универсальные подходы к структурированию ОС.

**Ядро и вспомогательные модули ОС**

Модули ОС делятся на две группы:

- ядро — модули, выполняющие основные функции ОС;

- модули, выполняющие вспомогательные функции ОС.

Модули ядра выполняют такие базовые функции ОС, как управление процессами, памятью, устройствами ввода-вывода и т. п. В состав ядра входят функции, решающие внутрисистемные задачи организации вычислительного процесса, такие как переключение контекстов, загрузка/выгрузка страниц, обработка прерываний. Эти функции недоступны для приложений.

Другой класс функций ядра служит для поддержки приложений, создавая для них так называемую прикладную программную среду. Приложения могут обращаться к ядру с запросами — системными вызовами — для выполнения тех или иных действий, например для открытия и чтения файла, вывода графической информации на дисплей, получения системного времени и т. д. Функции ядра, которые могут вызываться приложениями, образуют интерфейс прикладного программирования — API.

Функции, выполняемые модулями ядра, являются наиболее часто используемыми функциями ОС, поэтому скорость их выполнения определяет производительность всей системы в целом. Для обеспечения высокой скорости работы ОС все модули ядра или большая их часть постоянно находятся в оперативной памяти, то есть являются **резидентными**.

Ядро является движущей силой всех вычислительных процессов в компьютерной системе, и крах ядра равносилен краху всей системы.

Остальные модули ОС выполняют весьма полезные, но менее обязательные функции. Например, к таким вспомогательным модулям могут быть отнесены программы архивирования данных на магнитной ленте, дефрагментации диска, текстового редактора. Вспомогательные модули ОС оформляются либо в виде приложений, либо в виде библиотек процедур.

**Состав ядра** Аппарт. обесп. ЭВМ

1) Средства аппаратной поддержки

2) машино зависимые ядра

3) базовые механизмы ядра

4) менеджеры ресурсов

5) интерфейс системных выходов

6)API

***Вспомогательные модули ОС обычно подразделяются на следующие группы:***

- **утилиты** — программы, решающие отдельные задачи управления и сопровождения компьютерной системы, такие, например, как программы сжатия дисков, архивирования данных на магнитную ленту;

**- системные обрабатывающие программы** — текстовые или графические редакторы, компиляторы, компоновщики, отладчики;

**- программы предоставления пользователю дополнительных услуг** — специальный вариант пользовательского интерфейса, калькулятор и даже игры;

- **библиотеки процедур** различного назначения, упрощающие разработку приложений, например библиотека математических функций, функций ввода-вывода и т. д.

**Режимы работы ОС**

Для надежного управления ходом работы компьютера ОС должна иметь определенные привилегии по отношению к другим задачам (приложениям). Иначе некорректно работающее приложение может вмешаться в работу ОС и испортить часть ее кода.

Обеспечить привилегии ОС невозможно без специальных средств аппаратной поддержки. Аппаратура компьютера должна поддерживать, как минимум, два режима работы:

- **Пользовательский** режим (user mode)

- **Привилегированный** режим, режим ядра (kernel mode)

Так как ядро выполняет все основные функции ОС, то ядро работает в привилегированном режиме. Приложения работают в пользовательском режиме.

Приложение ставиться в подчиненное положение за счет запрета выполнения в пользовательском режиме команд, связанных с управлением ресурсами компьютера. Например, команда доступа к памяти для приложения разрешается, когда команда обращается к области памяти, отведенной данному приложению, и запрещается при обращении к областям памяти, занятых самой ОС или другими приложениями. Может быть создано несколько уровней привилегий, процессор Intel поддерживает четыре уровня.

Ядро и приложения могут работать в одном режиме, без поддержки привилегированного режима, например процессор Intel 8088/86, первый процессор для персональных компьютеров фирмы IBM. Некорректно написанные приложения для MS DOS могли разрушить модули самой ОС, что иногда и происходило, но область использования ОС MS DOS не предъявляла высоких требований к надежности.

Часть операционной системы, которая работает в режиме пользователя, так же называется **операционным окружением** (вспомогательные модули). Операционное окружение предназначено для связи приложений и самой операционной системы.

При выполнении приложений, как правило, требуется обращение к ОС, для выполнения определенных действий, например получения системного времени. Приложение обращается к функциям ОС с помощью системных вызовов. Таким образом, операционное окружение должно обеспечить:

1. Организацию диалога пользователя;

2. Выполнение системных вызовов;

3. Обращение к библиотечным функциям;

4. Вызов стандартных обслуживающих программ;

**7. Структура процессора**

Архитектура типичной небольшой вычислительной системы на основе микроЭВМ показана на рис. 2.1 Такая микроЭВМ содержит все 5 основных блоков цифровой машины: устройство ввода информации, управляющее устройство (УУ), арифметико-логическое устройство (АЛУ) (входящие в состав микропроцессора), запоминающие устройства (ЗУ) и устройство вывода информации.

Микропроцессор координирует работу всех устройств цифровой системы с помощью шины управления (ШУ). Помимо ШУ имеется 16-разрядная адресная шина (ША), которая служит для выбора определенной ячейки памяти, порта ввода или порта вывода. По 8-разрядной информационной шине или шине данных (ШД) осуществляется двунаправленная пересылка данных к микропроцессору и от микропроцессора.

Важно отметить, что МП может посылать информацию в память микроЭВМ или к одному из портов вывода, а также получать информацию из памяти или от одного из портов ввода.

Постоянное запоминающее устройство (ПЗУ) в микроЭВМ содержит некоторую программу (на практике программу инициализации ЭВМ). Программы могут быть загружены в запоминающее устройство с произвольной выборкой (ЗУПВ) и из внешнего запоминающего устройства (ВЗУ). Это программы пользователя.

МПС микропроцессор выполняет следующие функции:   
- выборку команд программы из основной памяти;   
- дешифрацию команд;   
- выполнение арифметических, логических и других операций, закодированных в командах;   
- управление пересылкой информации между регистрами и основной памятью, между устройствами ввода/вывода;   
- отработку сигналов от устройств ввода/вывода, в том числе реализацию прерываний с этих устройств;   
- управление и координацию работы основных узлов МП.

**19.Прерывания. Типы и классы прерываний.**

**Прерывание** представляет собой специальный сигнал идущий от устройства к процессу . Этот сигнал сообщает процессу о том что возникла ситуация требующая немедленной обработки.

Все прерывания в ОС имеют строго определённые программы называемых "обработчиками прерывания".

Всего существует 3 вида прерываний:

1) **Аппаратные** (возникает при нажатии клавиши на клавиатуре или курсоре или при падении напряжения), так же называют **внешние.** Возникает на границе машинных тактов.

2) **Логические** (исключение) ( возникает при попытки выполнить запрещённые действия, например деление на 0, переполнение регистров). Называется **внутренним.**

**3) Программное прерывание.** По сути прерыванием не является, поскольку не формируется аппаратный сигнал. Но в момент программного прерывания перезагружаются значения слова состояния и счётчики команд.

**Существует 6 классов прерываний.**

1. Программы-прерывания по системной директиве. Инициатор - активный процесс, выполнивший обращение к супервизору, то есть запрос на предоставление ОС системной вычислительной услуги.

2. Логические прерывания по контролю программы вызываются программными ошибками, обнаруженными при выполнении программы (деление на 0, попытка выполнить привилегированную команду в пользовательском режиме, неверный код операции, защита памяти и т.д.).

3. Прерывание по таймеру.

4. Прерывания ввода-вывода. Например: завершение операции, ошибка, переход устройства в состояние готовности.

**14.Средства аппаратной поддержки ОС.**

Практически все современные аппаратные платформы имеют некоторый типичный набор средств аппаратной поддержки ОС, в который входят следующие компоненты: 1)средства поддержки привилегированного режима; 2)средства трансляции адресов; 3)средства переключения процессов; 4)система прерываний; 5)системный таймер; 6)средства защиты областей памяти.  
***Средства поддержки привилегированного режима*** обычно основаны на системном регистре процессора, часто называемом «словом состояния» машины или процессора. В обязанности средств поддержки привилегированного режима входит выполнение проверки допустимости выполнения активной программой инструкций процессора при текущем уровне привилегированности.  
***Средства трансляции адресов*** выполняют операции преобразования виртуальных адресов, которые содержатся в кодах процесса, в адреса физической памяти.  
***Средства переключения процессов*** предназначены для быстрого сохранения контекста приостанавливаемого процесса и восстановления контекста процесса, который становится активным. Содержимое контекста обычно включает содержимое всех регистров общего назначения процессора, регистра флагов операций (то есть флагов нуля, переноса, переполнения и т. п.), а также тех системных регистров и указателей, которые связаны с отдельным процессом, а не операционной системой.  
***Система прерываний*** позволяет компьютеру реагировать на внешние события, синхронизировать выполнение процессов и работу устройств ввода-вывода, быстро переходить с одной программы на другую. Механизм прерываний нужен для того, чтобы оповестить процессор о возникновении в вычислительной системе некоторого непредсказуемого события или события, которое не синхронизировано с циклом работы процессора.  
***Системный таймер***, часто реализуемый в виде быстродействующего регистра-счетчика, необходим операционной системе для выдержки интервалов времени. После того, как время требуемого интервала истекает, таймер инициирует прерывание, которое обрабатывается процедурой операционной системы. Прерывания от системного таймера используются ОС в первую очередь для слежения за тем, как отдельные процессы расходуют время процессора.  
***Средства защиты областей памяти***обеспечивают на аппаратном уровне проверку возможности программного кода осуществлять с данными определенной области памяти такие операции, как чтение, запись или выполнение (при передачах управления).

**23.Управление памятью. Схема распределения памяти.**

Существует 2 группы методов распределения памяти:

1) Деление памяти на разделы

2) распределение памяти на концепции виртуализации

Концепции виртуализации предполагает ,что некоторый ресурс компьютера может быть представлен пользователи или прикладной программе.

Виртуальный ресурс представляет собой некую аппаратно-программную модель конкретного ресурса и в частности моделирования большого объекта памяти реализуется с помощью внешней памяти.

Если процесс должен быть создан в системе , то ему выделяется некая область ОП , для размещения образа процесса.

Если не хватает места, то старый процесс выгружается в swap область , и на его место записывается новый.

**Схема**

Методы распределения памяти:

1)Без использования внеш. памяти

1.С фиксированными разделами

2.С разделами переменного размера

3.С переменными разделами

2)С использование(на основе виртуализации)

1. Свокинг

2. Страничная распр.

3. Сегментное распр.

4. Сегментно-страничная распр.

**21.Слово состояния SW и его структура.**

**Регистр слова состояния SW**- содержит ряд признаков характеризующих состояние процессора на момент выполнения команды. **Признаки**: режим работы процессора в момент прохождения процесса проходит перезагрузку счётчика команд PC. PC- содержит в себе адрес ячейки ОП в которой содержится команда которая должна будет выполнена в следующий машинный такт.

Существует 6 разрядов состояния:

1) i -недопустимая операция

2) d- денормализованный операнд

3) z- деления на ноль

4) o- переполнение

5) u- слишком малый результат

6) p- потеря точности

**15.Основные типы организации многозадачности и типы многозадачных ОС.**

**Многозадачность**— свойство операционной системы обеспечивать возможность параллельной (или псевдопараллельной) обработки нескольких процессов. Истинная многозадачность операционной системы возможна только в распределённых вычислительных системах.

Многозадачные системы позволяют запустить одновременно несколько программ, которые будут работать параллельно.

Существует 2 типа многозадачности:

- **Процессная многозадачность** (основанная на процессах — одновременно выполняющихся программах). Здесь программа — наименьший элемент кода, которым может управлять планировщик операционной системы. Более известна большинству пользователей (работа в текстовом редакторе и прослушивание музыки).

- **Поточная многозадачность** (основанная на потоках). Наименьший элемент управляемого кода — поток (одна программа может выполнять 2 и более задачи одновременно).

**Многопоточность** — специализированная форма многозадачности.

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

- однозадачные (MS-DOS, MSX);

- многозадачные (ОС ЕС, OS/2, UNIX, Windows 95).

Однозадачные ОС в основном выполняют функцию представления пользователю виртуальной машины, делая более простым и удобным процесс взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.

Многозадачные ОС, кроме вышеперечисленных функций, управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы и внешние устройства.

Поддержка многопользовательского режима. По числу одновременно работающих пользователей ОС делятся на:

- **однопользовательские** (MS-DOS, Windows3.x, ранние версии OS/2);

- **многопользовательские** (UNIX, Windows NT).

Главным отличием многопользовательских систем от однопользовательских является наличие средств защиты информации каждого пользователя от несанкционированного доступа других пользователей. Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

Многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса в системе с многопроцессорной архитектурой: асимметричные ОС и симметричные ОС. Асимметричная ОС целиком выполняется только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам. Симметричная ОС полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

- системы пакетной обработки (например, OC EC),

- системы разделения времени (UNIX, VMS),

- системы реального времени (QNX, RT/11).

**22.Преоритетность и маскирование прерываний.**

***Маскирование***

Внешние прерывания в зависимости от возможности запрета делятся на:

- **маскируемые** — прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний

- **немаскируемые**— обрабатываются всегда, независимо от запретов на другие прерывания.

Запрет на прерывание класса обусловлен тем, если в какой то момент обработки прерывания какого то типа вдруг произойдёт прерывание такого же типа, то произойдёт затирание старых значений новыми.

***Приоритизация*** : 1) SVC 2)лог. 3) таймер 4) ввод/вывод

Такие расположение приоритетов обусловлены тем ,что более длительное прерывание может тормозить более быстрые.

Прерывания запрещённые по маске являются временно отложенными, поскольку сигнал воспринимается и назначается обработкой прерывания.

**13.Многослойная структура ядра ОС.**

Ядро может состоять из следующих слоев:  
***Средства аппаратной поддержки ОС****.* Часть функций ОС может выполняться аппаратными средствами. Поэтому иногда можно встретить определение операционной системы как совокупности программных и аппаратных средств. К операционной системе относят только средства аппаратной поддержки ОС, то есть те, которые прямо участвуют в организации вычислительных процессов: средства поддержки привилегированного режима, систему прерываний, средства переключения контекстов процессов, средства защиты областей памяти и т. п.  
***Машинно-зависимые компоненты ОС****.* Этот слой образуют программные модули, в которых отражается специфика аппаратной платформы компьютера. В идеале этот слой полностью экранирует вышележащие слои ядра от особенностей аппаратуры. Это позволяет разрабатывать вышележащие слои на основе машинно-независимых модулей, существующих в единственном экземпляре для всех типов аппаратных платформ, поддерживаемых данной ОС.  
***Базовые механизмы ядра****.* Этот слой выполняет наиболее примитивные операции ядра, такие как программное переключение контекстов процессов, диспетчеризацию прерываний, перемещение страниц из памяти на диск и обратно и т. п. Модули данного слоя не принимают решений о распределении ресурсов - они только отрабатывают принятые «наверху» решения, что и дает повод называть их исполнительными механизмами для модулей верхних слоев.  
***Менеджеры ресурсов****.* Этот слой состоит из мощных функциональных модулей, реализующих стратегические задачи по управлению основными ресурсами вычислительной системы. Обычно на данном слое работают менеджеры (называемые также диспетчерами) процессов, ввода-вывода, файловой системы и оперативной памяти. Разбиение на менеджеры может быть и другим.  
***Интерфейс системных вызовов****.* Этот слой является самым верхним слоем ядра и взаимодействует непосредственно с приложениями и системными утилитами, образуя прикладной программный интерфейс операционной системы. Функции API, обслуживающие системные вызовы, предоставляют доступ к ресурсам системы в удобной и компактной форме, без указания деталей их физического расположения.

**20.Механизм обработки прерываний.**

Механизм обработки прерываний включает в себя несколько этапов:

1) Сохранение специальной рабочей области прерывания значений слова состояния SW и счётчики команд для прерываемого процесса. Для того что бы по окончанию обработать прерывания можно было бы восстановить работу процесса с прерванного места.

2) Загрузка в регистры SW и PC значений соответствующих конкретному обработчику прерывания и передачи управления этому обработчику.

3) Выполнение процедуры обработки прерывания.

4) По окончанию обработки ,вызов диспетчера задач, который может активизировать прерванный процесс, если в очереди готовности к процессору нет более приоритетных процессов

**24.Организация Виртуальной памяти. Свопинг. Разделение сегментов памяти**

Для экономии ОП в выделенную область загружается не все команды и не весь образ процесса , а остальные части подгружаются по мере необходимости с внешней памяти в оперативную.

Таким образом ОП моделируется за счёт внешней. (Виртуализация памяти).

**Виртуа́льная па́мять**— технология управления памятью ЭВМ, разработанная для многозадачных операционных систем. При использовании данной технологии для каждой программы используются независимые схемы адресации памяти, отображающиеся тем или иным способом на физические адреса в памяти ЭВМ.

Применение механизма виртуальной памяти позволяет:

- упростить адресацию памяти клиентским программным обеспечением;

- рационально управлять оперативной памятью компьютера (хранить в ней только активно используемые области памяти);

- изолировать процессы друг от друга (процесс полагает, что монопольно владеет всей памятью).

Виртуальная память имеет сегментно-страничную организацию и реализована в иерархической системе памяти ЭВМ.

***Подкачка страниц (swapping)*** — один из механизмов виртуальной памяти, при котором отдельные фрагменты памяти (обычно неактивные) перемещаются из ОЗУ на жёсткий, освобождая ОЗУ для загрузки других фрагментов памяти(страницы памяти).

Разделяемые сегменты памяти- области памяти которым могут обращаться совместно несколько процессов.

В этом случае разделение сегмента присутствует во всех образах работающих с ним процессы, но в физ. памяти он один.

**26. Файловая структура диска. Стартовый сектор и FAT-таблица.**

**Файловая система (ФС)** - функциональная часть ОС, т.е. это порядок хранения и - организации файлов на диске

**Виды файловой структуры**:

1) Одноуровневая ФС - линейная последовательность имен файлов, используется для дисков с небольшим количеством файлов;

2) Многоуровневая иерархическая ФС - представляет собой древовидную структуру, служит для хранения сотни и тысячи файлов.

Каталог (Папка) верхнего уровня содержит вложенные папки 1уровня, которые могут содержать папки 2 уровня и т.д.

Для хранения информации каждый диск разбивается на 2 области: 1) каталог (directory) или папка - содержит названия файлов и указание на начало их размещения на диске; 2) область хранения файлов, содержит текст.

Чтобы найти файл надо знать: 1)имя файла; 2) где храниться файл

Элементы файловой структуры :

- стартовый сектор(сектор начальной загрузки, Boot-сектор),

- таблица размещения файлов (FAT - File Allocation Table),

- корневой каталог(Root-Directory),

- область данных(оставшееся свободным дисковое пространство)

Эти элементы создаются специальными программами (в среде MS DOS) в процессе инициализации диска.

***Стартовый сектор(сектор начальной загрузки,Boot-сектор) :***

Здесь записана информация, необходимая MS DOS для работы с диском :

- идентификатор OS(если диск системный),

- размер сектора диска,

- количество секторов в кластере,

- количество резервных секторов в начале диска,

- количество копий FAT на диске(стандарт - две),

- количество элементов в каталоге,

- количество секторов на диске,

- тип формата диска,

- количество секторов в FAT,

- количество секторов на дорожку,

- количество поверхностей,

- блок начальной загрузки OS,

За стартовым сектором располагается FAT.

FAT(таблица размещения файлов) :

Область данных диска(см.выше) представлена в MS DOS как последовательность пронумерованных кластеров.

**FAT** - это массив элементов, адресующих кластеры области данных диска.

Каждому кластеру области данных соответствует один элемент FAT.

Элементы FAT служат в качестве цепочки ссылок на кластеры файла в области данных.

**FAT** - крайне важный элемент Файловой структуры. Нарушения в FAT могут привести к полной или частичной потери информации на всем логическом диске. Именно поэтому, на диске хранится две копии FAT. Существуют спец.программы, которые контролируют состояние FAT и исправляют нарушения.

**Корневой каталог :**Это определенная область диска, создаваемая в процессе инициализации(форматировании) диска, где содержится информация о файлах и каталогах, хранящихся на диске.

Корневой Каталог всегда существует на отформатированном диске. На одном диске всегда бывает только один корневой каталог. Размер корневого каталога для данного диска - величина фиксированная, поэтому максимальное количество "привязанных" к нему файлов и других (дочерних) каталогов (Подкаталогов) - строго определенное.

MS-DOS - 16-разрядная операционная система, работающая в реальном режиме процессора.

**30.Файловые системы s5 и ufs (UNIX).**

Вместо термина «кластер» будет использоваться термин «блок», как это принято в файловых системах UNIX.

Файловые системы s5 (получившие название от System V) и ufs (UNIX File System) используют очень близкую физическую модель. Система ufs является развитием системы s5. Файловая система ufs расширяет возможности s5 по поддержке больших дисков и файлов, а также повышает ее надежность.

Раздел диска, где размещается файловая система, делится на четыре области:

**загрузочный блок;**

**суперблок (superblock**) -содержит самую общую информацию о файловой системе: размер файловой системы, размер области индексных дескрипторов, число индексных дескрипторов, список свободных блоков и список свободных индексных дескрипторов, а также другую административную информацию;

**область индексных дескрипторов (inode list**) – содержит набор индексных дескрипторов, перенумерованных в порядке расположения их номеров;

**область данных,** в которой расположены как обычные файлы, так и файлы-каталоги, в том числе и корневой каталог; специальные файлы представлены в файловой системе только записями в соответствующих каталогах и индексными дескрипторами специального формата, но места в области данных не занимают.

**Физическая организация файловой системы ufs**

В этих повторяющихся последовательностях областей, суперблок является резервной копией основной первой копии суперблока. При повреждении основной копии суперблока может быть использована резервная копия суперблока. Области же блока группы цилиндров и индексных дескрипторов содержат индивидуальные для каждой последовательности значения. Блок группы цилиндров описывает количество индексных дескрипторов и блоков данных, расположенных на данной группе цилиндров диска. Такая группировка делается для ускорения доступа, чтобы просмотр индексных дескрипторов и данных файлов, описываемых этими дескрипторами, не приводил к слишком большим перемещениям головок диска.

Кроме того, в ufs имена файлов могут иметь длину до 255 символов (кодировка ASCII, по одному байту на символ), в то время как в s5 длина имени не может превышать 14 символов.