БИЛЕТ № 12

1) Процедура начальной загрузки ОС.

Вначале происходит тестирование системы – определяются параметры системы.
Загрузка с BIOS – программа начальной загрузки – загружает частично таблицу векторов прерываний в ОП, и переписывает программы для этой таблицы. Загрузка с Boot-сектора бутового загрузчика, который определяет активный раздел в соответствии находится начало активного раздела. Бутовый загрузчик находится в MBR. По адресу 0 0 0 находится jump на программу начальной загрузки в активном разделе. Загрузка ОС(инициализация) происходит в два этапа: init-ядра, init-sys
Когда синициализировали систему, подгружаем autoexec и command.com

2) Система управления памятью. Решаемые задачи.

часть операционной системы, отвечающая за упр. пам. называется модуль управления памятью (менеджер памяти). Его функции: учет свободных и занятых участков памяти, выделение и освобождение памяти процессам, управление обменом данных между ОП и диском, если памяти недостаточно.

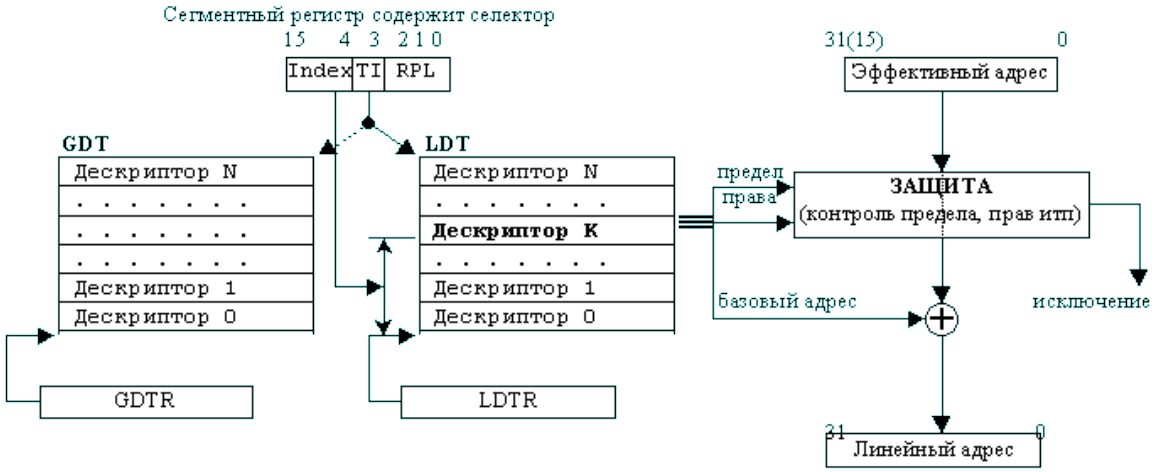
3) Особенности управления внешней памятью в системах UNIX.

4) Состав системных объектов. Реализация защищенного режима.

Ответ:

Доп.инфа:

Особенности адресации в защищенном режиме.



TI (Table Indicator) – выбирает дескрипторную таблицу (TI=0 : GDT; TI=1 : LDT)
RPL (Requested Privilege Level) – запрашиваемый уровень привилегий

Дескриптор - это 8-байтная единица описательной информации, распознаваемая устройством управления памятью в защищенном режиме, хранящаяся в дескрипторной таблице. Дескриптор сегмента содержит базовый адрес описываемого сегмента, предел сегмента и права доступа к сегменту. Дескрипторы являются основой защиты и мультизадачности. В защищенном режиме сегменты могут начинаться с любого линейного адреса и иметь любой предел вплоть до 4Гбайт. Существуют две обязательных дескрипторных таблицы - глобальная (GDT) и дескрипторная таблица прерывания (IDT),- а также множество (до 8192) локальных дескрипторных таблиц (LDT), из которых в один момент времени процессору доступна только одна. Дескрипторы сегментов могут находиться в GDT или LDT. Расположение дескрипторных таблиц определяется регистрами процессора GDTR, IDTR, LDTR. Регистры GDTR и IDTR - 6-байтные, они содержат 32 бита линейного базового адреса дескрипторной таблицы и 16 бит предела таблицы. Программно доступная часть регистра LDTR - 16 бит, которые являются селектором LDT. Дескрипторы LDT находятся в GDT. Однако чтобы не обращаться каждый раз к GDT в процессоре имеется теневая (программно недоступная) часть регистра LDTR, в которую процессор помещает дескриптор LDT при каждой перегрузке селектора в регистре LDTR.

Значения, помещаемые в сегментные регистры, называются селекторами. Селектор содержит индекс дескриптора в дескрипторной таблице, бит определяющий, к какой дескрипторной таблице производится обращение (LDT или GDT), а также запрашиваемые права доступа к сегменту. Таким образом, селектор выбирает дескрипторную таблицу, выбирает дескриптор из таблицы, а по дескриптору определяется положение сегмента в линейном пространстве памяти. Однако обращение к дескрипторным таблицам происходит только при загрузке селектора в сегментный регистр. При этом процессор помещает дескриптор в теневую (программно недоступную) часть сегментного регистра. При формировании линейного адреса дескриптор сегмента процессору уже известен.

5) Сокеты, особенности их использования.

Soket – новая точка коммутации

КОНСПЕКТ тоже…

Сокеты – это средства межпроцессорного взаимодействия между процессорами разных вычислительных систем

Для обозначения коммуникационного узла, обеспечивающего прием и передачу данных для объекта (процесса), был предложен специальный объект — *сокет* (socket). Сокеты создаются в рамках определенного коммуникационного домена, подобно тому, как файлы создаются в рамках файловой системы. Сокеты имеют соответствующий интерфейс доступа в файловой системе, и так же как обычные файлы, адресуются некоторым целым числом — дескриптором. Однако в отличие от обычных файлов, сокеты представляют собой виртуальный объект, который существует, пока на него ссылается хотя бы один из процессов.

- *Сокет датаграмм* (datagram socket), через который осуществляется теоретически ненадежная, несвязная передача пакетов.
- *Сокет потока* (stream socket), через который осуществляется надежная передача потока байтов без сохранения границ сообщений. Этот тип сокетов поддерживает передачу экстренных данных.
- *Сокет пакетов* (packet socket), через который осуществляется надежная последовательная передача данных без дублирования с предварительным установлением связи. При этом сохраняются границы сообщений.
- *Сокет низкого уровня (raw* socket), через который осуществляется непосредственный доступ к коммуникационному протоколу.

Наконец, для того чтобы независимые процессы имели возможность взаимодействовать друг с другом, для сокетов должно быть определено *пространство имен.* Имя сокета имеет смысл только в рамках коммуникационного домена, в котором он создан.

Примитивы сокетов для TCP протокола:

- SOCKET – создать точку коммуникации
- BIND – назначить сокету локальный адрес
- LISTEN – обозначить готовность к установке соединения
- ACCEPT – заблокировать вызывающую сторону до прибытия запроса на соединение
- CONNECT – попытка установить соединение
- SEND, WRITETO, SENDTO – послать сообщение сокету в зависимости от типа сокета и сообщения
- RECEIVE, READ, RECIEVEFROM – принять сообщение

CLOSE – разорвать соединение

Коммуникационный канал создаётся при создании сокета между источником и получателем и имеет такие характеристики:

- Коммуникационный протокол
- Локальный адрес источника
- Локальный адрес процесса источника
- Удалённый адрес получателя

Удалённый адрес процесса получателя