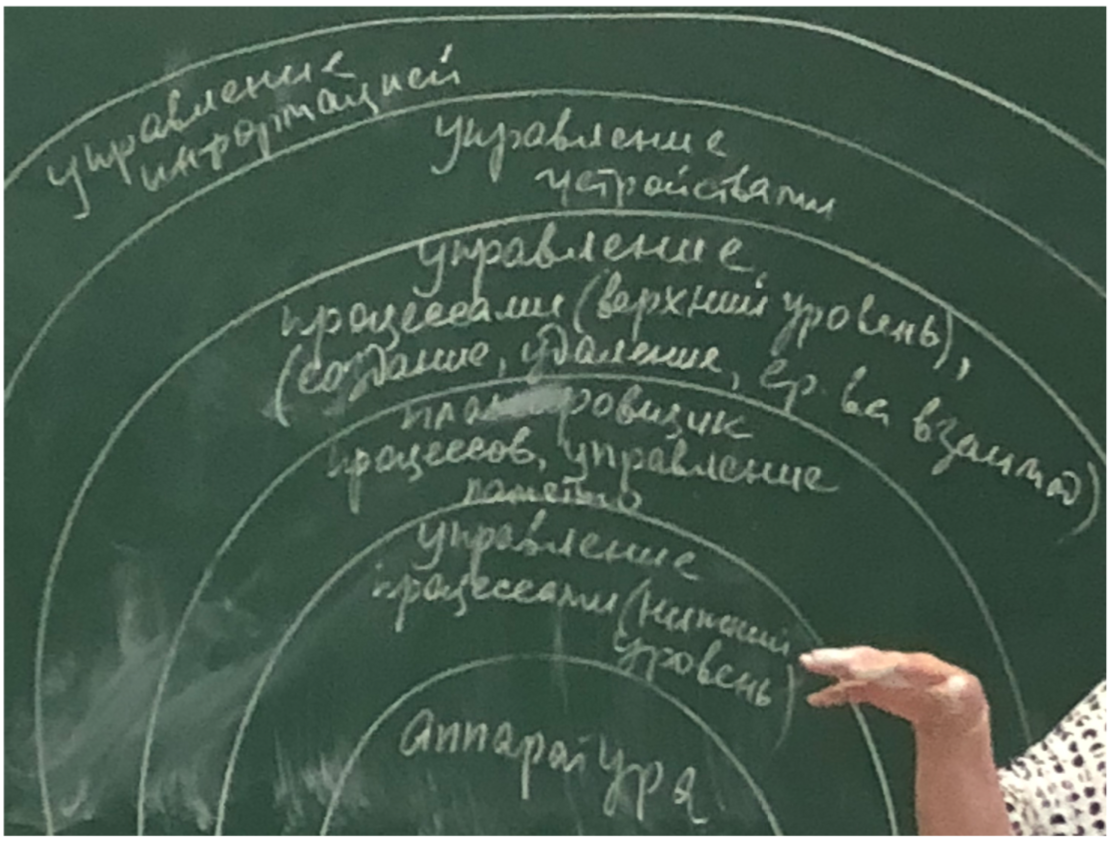
# Операционные системы Лекция (27.12.2021)



Структура ядра системы: Программное обеспечение ставится на голое железо, на аппаратную часть.

* нижний уровень - аппаратная часть
* уровень 1 - управление процессорами (нижний уровень; диспетчеризация - выделение процессу процессорного времени);
* уровень 2 - (P, V) - семафорные операции, контр. доступа к разд. рес, планировщик процессов (операция более выс. уровня. постановка процессов в очередь к процессору или другим ресурсам), управление памятью; Процессор, оперативка, внешние устройства - аппаратные устройства
* уровень 3 - управление процессорами (верхний уровень -- создание и удаление процессов. Система принимает решение об этом, более высокоур. операция);
* уровень 4 - управление устройствами (подсистема ввода/вывода);
* уровень 5 - управление информацией (файловая система). Система предоставляет такие средства как наименование файлов, построения дерева катологов, удобных для пользователя.

Виртуальная машина - совокупность команд машины и команд ОС, которые могут использовать программы для получения сервиса ОС.

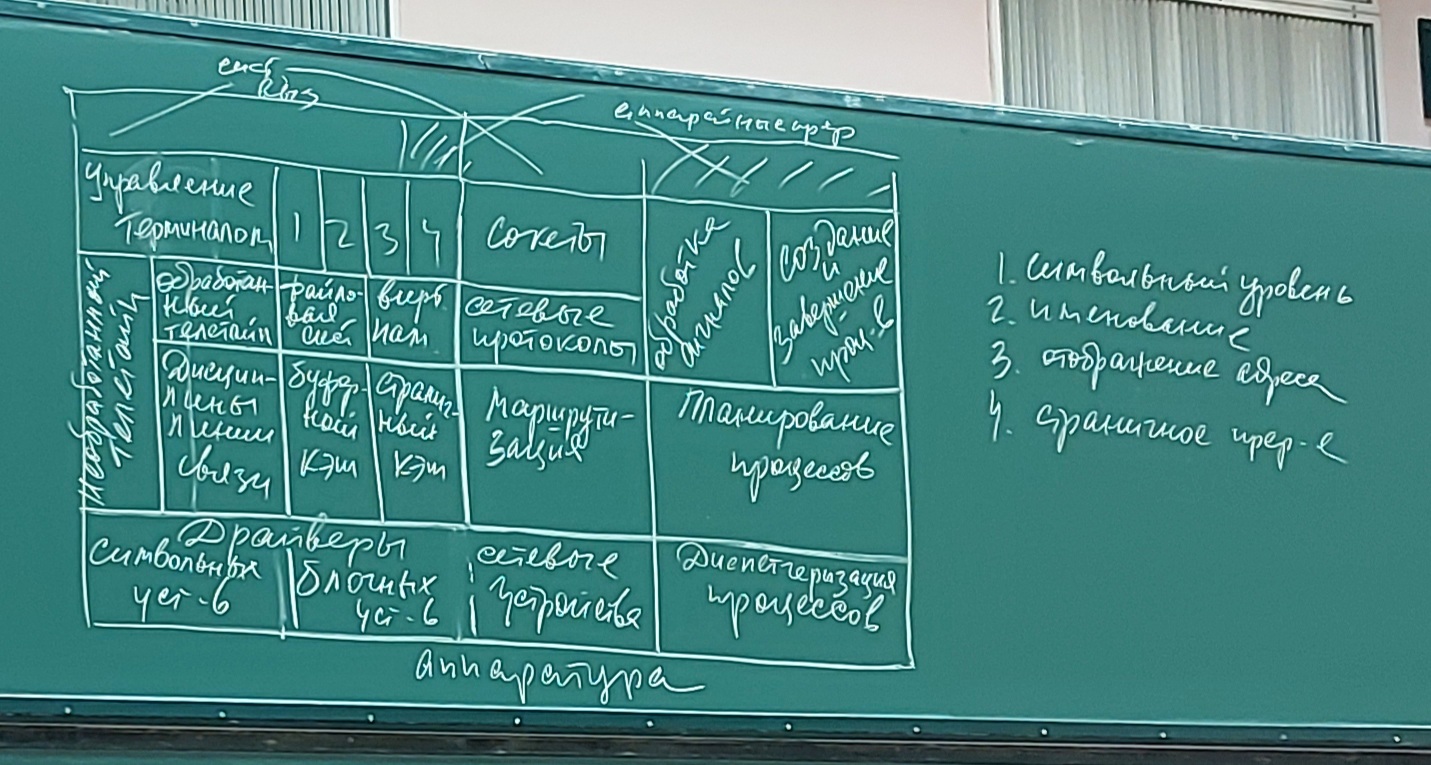
### Новая лекция

Интерфейс – набор функций, которые могут вызываться.

* Непрозрачный.  
  Функции верхнего только к следующему уровню
* Прозрачный.  
  Функции могут иметь доступ через уровень.
* Полупрозрачный.  
  Часть функций прозрачные, а часть непрозрачный.

## ****Иерархическая структура Unix BSD****





Диспетчеризация – это выделение процессам процессорного времени.

Планирование – определение в какой последовательности будут выполняться процессы.

Еще выше – создание и завершение процессов. (Создание – идентификация… Завершение – все ресурсы возвращаются системе)

(Сетевые устройства – символьные устройства)

Терминал – это клавиатура и монитор.

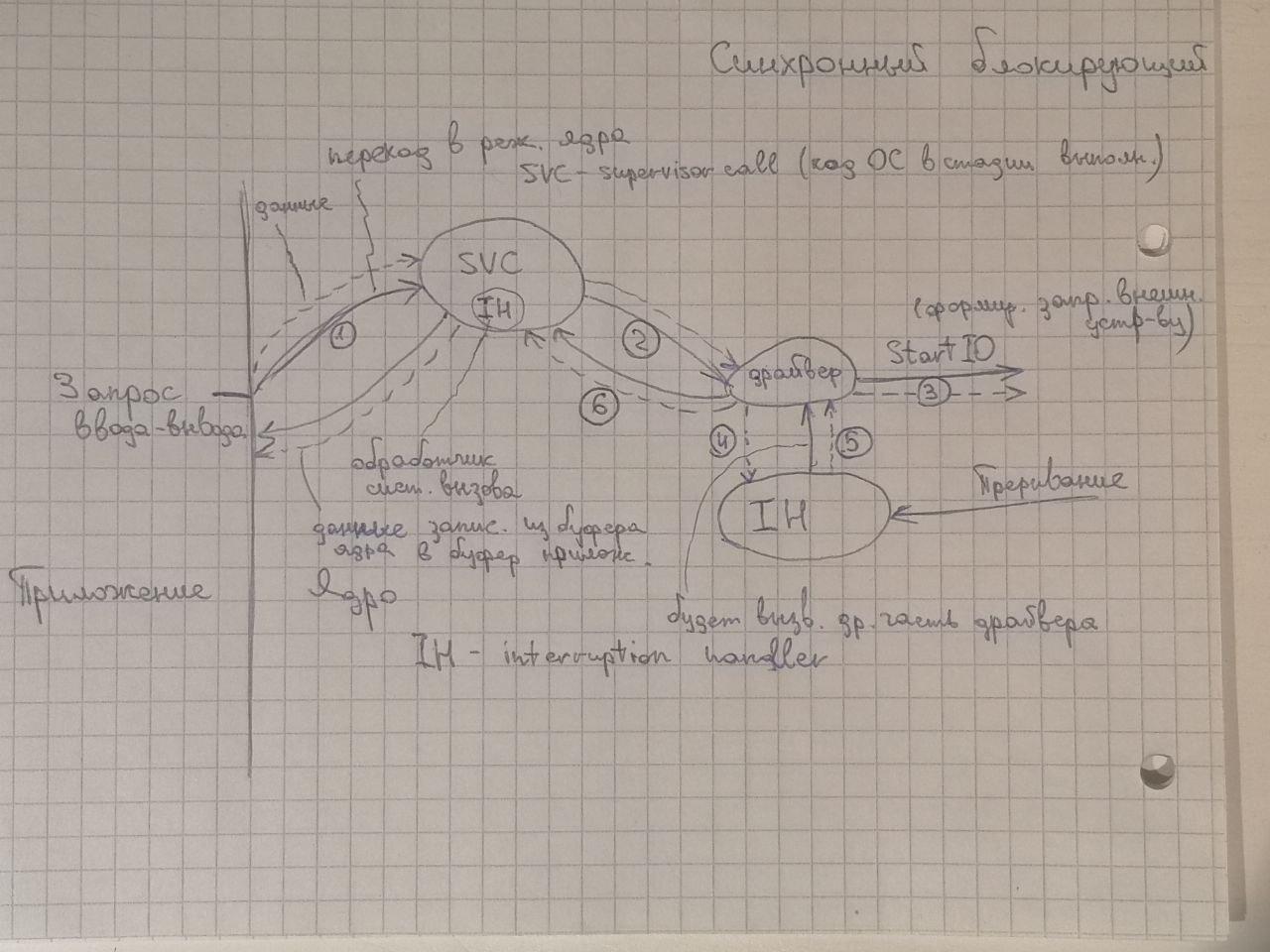
На этой иерархии и строится микро-ядерная структура. Все высокоуровневые процессы реализуются в виде самостоятельных функций (программ).

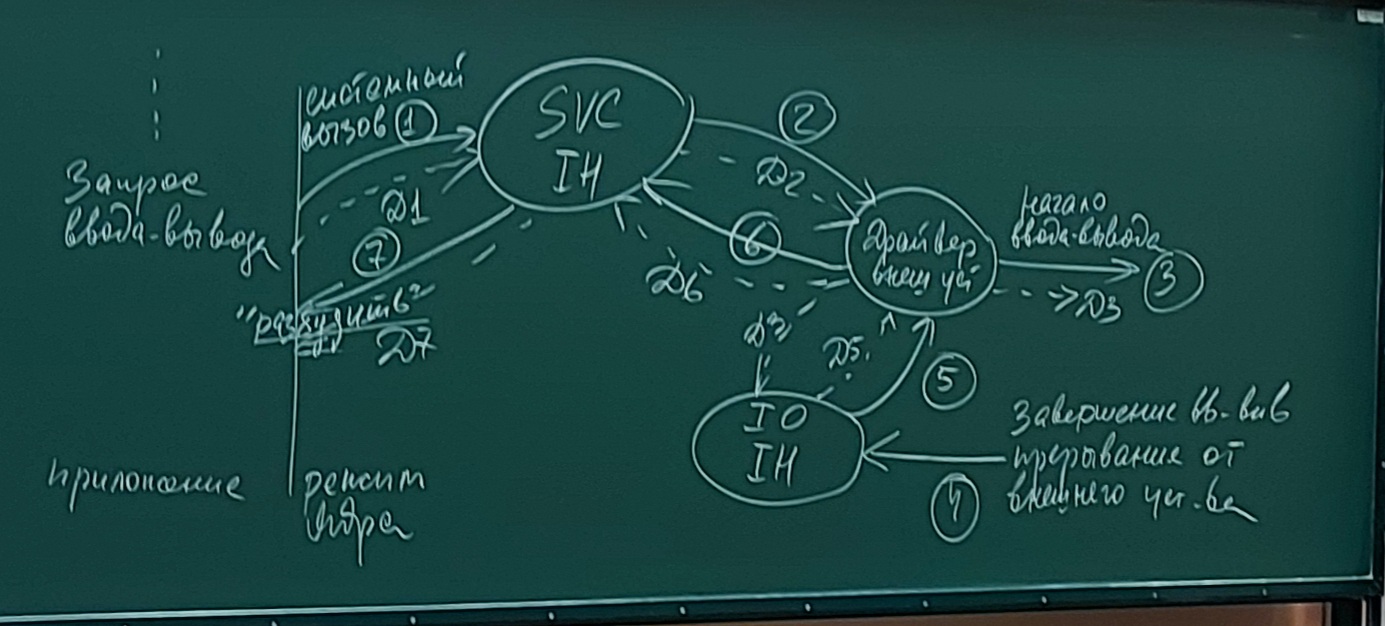
Мы обсуждали прерывания:

1. Системные вызовы (System call)
2. Исключения (Исключительные ситуации)
3. Аппаратные прерывания (Interrupts)

### Взаимодействие системы при запросе на ввод вывод

(Блокирующий синхронный ввод-вывод)





SVC – SuperVisor Call

Супервизор – это код ОС в стадии выполнения.

IH – Interrupt Handler

Драйвер внешнего устройства – программа ввода-вывода.

Обработчики прерываний входят в состав драйвера. Драйвер внешнего устройства всегда имеет один обработчик прерываний. (Обработчик драйвера – его точка входа)

Чтобы процесс продолжил выполняться его нужно “разбудить” (разблокировать – чтобы вернуть запрашиваемое значение).



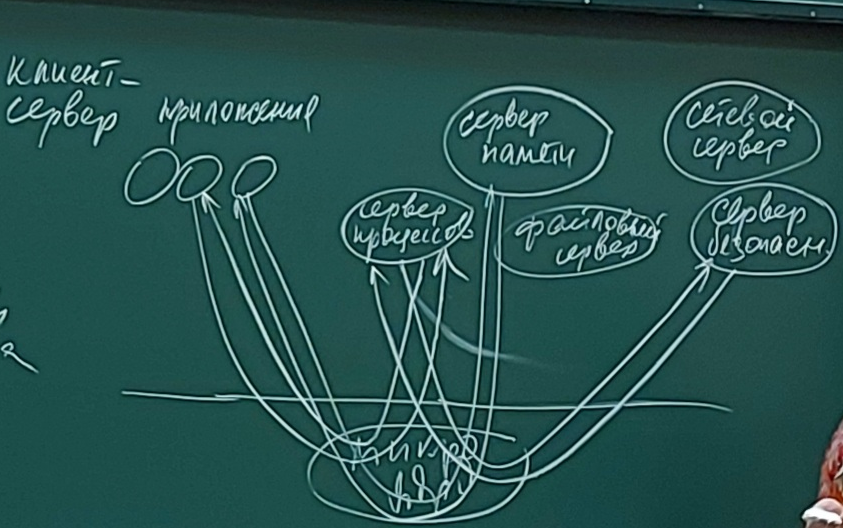
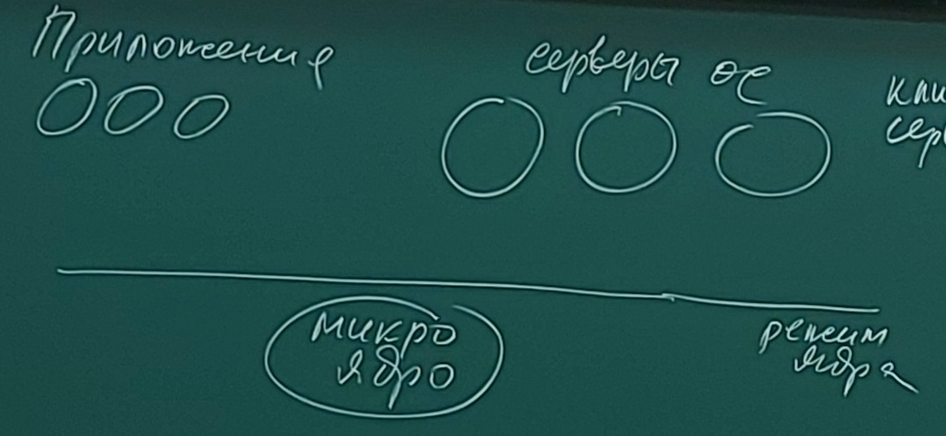
### Микро-ядерная архитектура

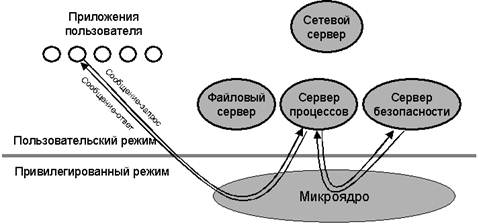
Микроядро (архитектура, появившаяся позже), которое известно также как клиент-серверные операционные системы и системы с обменом сообщениями; к этому классу относятся, например: QNX, MINIX 3, HURD, ядро Darvin MacOS, семейство ядер L4.

…

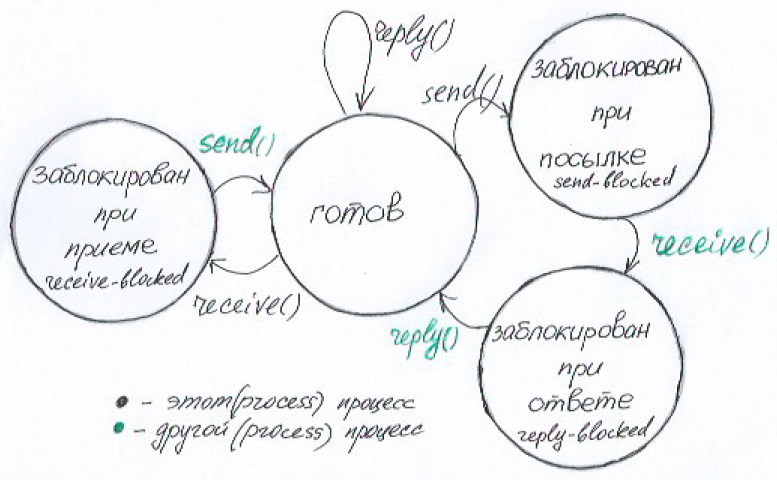
Программы ОС принято называть серверами ОС.

Клиент-серверная модель. Взаимодействие по некоторому протоколу. Протокол – это соглашение как о том как должно выполняться взаимодействие.



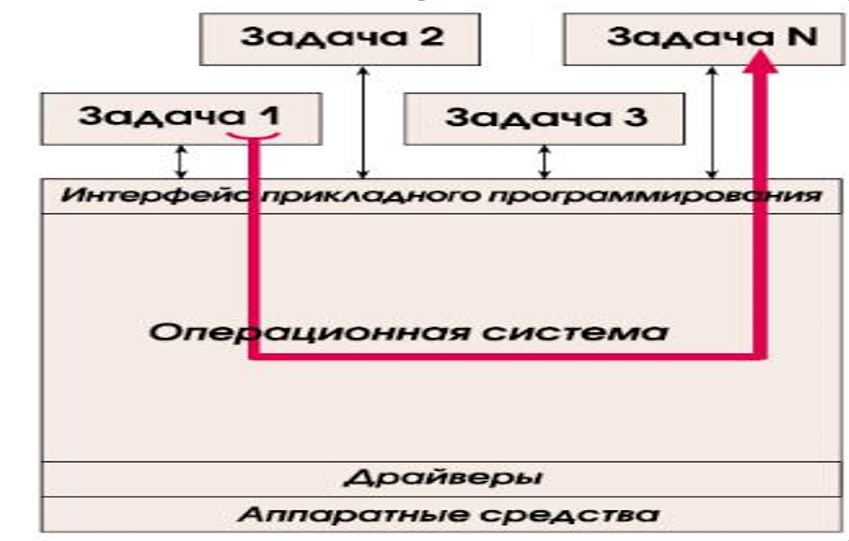


См диаграмму блокировки процессов при передаче сообщений (блокирован при посылке, блокирован при ответе, блокирован при приёме)



Процесс блокируется при посылке, если он вызвал send(), а процесс, которому сообщение адресовано, не готов к его приему. Как только адресат вызовет receive(), процесс, пославший сообщение будет, процесс, пославший сообщение будет заблокирован при ответе и будет в этом состоянии до тех пор пока запрос не будет обработан и на стороне другого процессе не будет вызван reply(). Если процесс вызвал receive(), а сообщение ему еще не послано, то процесс окажется заблокированным при приеме. Аналогичная ситуация будет при вызове reply().

### В монолитном ядре

В архитектуре с **монолитным ядром** все услуги для прикладного приложения выполняют отдельные части кода ядра (в адресном пространстве ядра) **

Монолитное ядро — это один процесс, работающий в одном адресном пространстве в привилегированном режиме. Представляет статический двоичный файл. Все службы ядра существуют и выполняются в адресном пространстве ядра.

### Микро-ядро Mach

Привлекательность микро-ядра: большая часть вынесена в пользов. … и может быть изменена, но ?микро-ядро Mach не такое уж микро?…

Но микро-ядерная архитектура используется в системах реального времени.

POSIX определение реального времени в ОС: Стандарт POSIX 10... .1 даёт определение – реальное время в ОС это способность ОС обеспечить требуемый уровень сервиса за определённый промежуток времени.

Задачи реального времени составляют одну из самых областей применение вычислительной техники. Это системы специального назначения (а у нас общего назначения).

Ядро ОС Mach оперирует пятью основными абстракциями:

* Task  
   – Базовая единица распределения ресурсов. Классическое представление о процессе как о владельце ресурсов  
  – Виртуальное адресное пространство Virtual address space, communication capabilities
* Thread  
   – Базовая единица диспетчеризации
* Port  
   – Коммуникационный канал для IPC (Inter Process Comnucations – Межпроцессное взаимодействие). Безопасные симплексные каналы связи, доступные только через функции отправки и получения (известные как права порта).
* Message  
   – Перемещается между программами через порт. Может содержать данные, определяемые возможностями порта, указатели.
* Memory Object   
  – Внутренние блоки управления памятью. Объекты памяти включают именованные записи и регионы; они представляют собой потенциально персистентные данные, которые могут отображаться в адресные пространства.