**Лектор**: Роман Тимурович

**ОС** – ПО, которая:

1) работает на вычислительной машине и управляет вычислительной машиной;  
2) ОС управляет вычислительным процессом (над ОС работают прикладные программы и ОС управляет, когда какая программа работает);  
3) ОС создает интерфейс для работы с железом.

ОС прослойка между железом (hardware) и софтом(software)  
ОС делится на ядро(kernel) и userspace.

Под ядром ОС подразумевают

Ядро ОС работает в более привилегированном режиме, чем userspace.

Интерфейс системных вызовов (system call interface) – связующая часть между ядром ОС и userspace.

Посикс (POSIX) – стандарт, который формализует системные вызовы к ОС. Нужен для переносимости кода (написали под линукс, можно использовать и под винду). Примеры где используется POSIX – OpenSolaris.

**Основные компоненты ОС:**

1) Драйвера и HAL (Hardware Abstraction Layer) –

2) Memory manager –

3) Планировщик – определяет какой процесс в какой момент времени будет исполняться, для этого у него должна быть информация о текущем выполняемом процессе – это process management (его можно выделить как отдельную компоненту)

4) Интерфейс

**Архитектура ОС**

Два подхода: микроядерный (ядро как можно меньше, как можно больше функций в userspace, это делается чтобы код был надежнее (чем меньше, тем надежнее)) и монолитный (все функции ОС находятся в ядре, ядро – единая большая программа).

Единственная успешная надежная микроядерная ОС QNX. Примеры микроядерных ОС: PIKC OS, SEL4, Kaspersky OS (10 тысяч строк кода – легко проверить и верифицировать).

Вынося функционал в userspace нужно использовать системные вызовы к ядру, на что тратится много времени. В связи с этим микроядерные ОС гораздо медленнее монолитных.

В монолитной ОС все что нужно находится в ядре ОС в едином адресном пространстве, следовательно, не нужно тратить время на переключения контекста процесса.

OS Windows - гибридная (есть драйвера и в ядре, и в userspace).

В монолитных проблема - безопасность, без перекомпиляции ядра невозможно увидеть новую вставленную мышку, так как нет нужного драйвера в едином адресном пространстве.

Ядро ОС винды и линукс – 30-40 тыс строк кода.

Примеры монолитных ОС: все линукс подобные (GNU – GNU Not Linux – рекурсивное определение),

Нет компаний, которые разрабатывают линукс, при разработке ядра линукса участвуют около 1000 инженеров из самых лучших компаний.

Android – это линукс ядро и реализация Java Machine от компании Google.  
Ядро Android – дистрибутив линукса.

ОС MobLin. Meego (Nokia N9) –> Tizen OS (телевизоры самсунг, цифровые часы) + Sailfish OS (считается российской ОС).

Система (жесткого) реального времени – это не про быстроту, это про гарантированные сроки выполнения. Это когда считаешь и не успеваешь досчитать, то уже не нужно досчитывать.

Система мягкого реального времени (некоторые разработчики не признают этого термина) – эта же задача может выполнять быстрее, но гарантия этого не обеспечена.

Линукс не является ОС реального времени, но под …

Винда – создатели утверждают, что это ОС жесткого реального времени.

Драйвер – компонент ОС, который управляет неким внешним устройством (аппаратурой) и предоставляет интерфейс для работы с этим устройством программам и прикладным процессам.

Драйвер должен предоставлять некий интерфейс, чтобы программы могли использовать некоторое устройство.

Процесс – программа на этапе исполнения.

Состояния процесса:

1) старт

2) готов к исполнению

3) исполняется (когда процесс находится на процессоре)

4) завершение процесса (когда он сам закончился)

5) остановка процесса (когда процесс снимается с процессора планировщиком) и его перевод в состояние «готов к исполнению»

6) процесс может вызвать блокировку, тогда процесс переходит в состояние «заблокирован» (например, ввод/вывод)

Загрузка ОС (линукс):

Подается питание на процессор X86 (ЦПУ). Первый программный код, который вступает в работу это биос (находится в ПЗУ) – он отвечает за …. У процессора есть РАБ

БИОС загружает и выполняет файл загрузки(MBR) (винда – …, линукс – граб). Потом передается управление от БИОС к загрузчику, который находит ядро, находит параметры, с которыми он должен загрузить ядро, копирует ядро в ОП и дальше загрузчик передает управление (команда джамп) коду ядра. Ядро запускает все драйвера.

Далее монтирование корневой файловой системы. В юзерлэнд находятся программы, которые исполняются в юзерспэйс. Виртуальная файловая система – разные памяти объединяются в единое адресное файловое пространство. Корневая файловая система – там находится самые главные программы, которые стартует самой первой.

Порты ввода/вывода

Относительно ЦПУ могут быть внешние устройства, которыми должна руководить ОС.

Обеспечивают взаимодействие порты ввода вывода и память ввода вывода.

Память ввода вывода: Часть адресов внешнего адресного пространства попадает…

Порт ввода вывода: то же самое, только через специальное устройство с инструкцией процессора.

Порты и память ввода вывода нужны для … чего-то там =(

Аппаратные внешние прерывания – способ как сказать внешнему устройству что что-то случилось. Изменяется сигнал и приходит на контроллер прерывания, который по коду прерывания понимает, что это за прерывание. Обработчик прерывания (код, который регистрирует ОС для обработки прерывания) не находится под управлением ОС.

Fork (вилка) – системный вызов – копирует один процесс в другое место (из родителя копируется потомок, они равноправны, но можно определить кто родитель, а кто потомок (возвращается разный код)), нужен чтобы клонировать процессы (есть такой вирус).

Exec (замещение) – системный вызов – замещение процессов, не создает новые процессы.

Сист вызовы Read, Write – посмотреть в красной книге.

Межпроцессорное взаимодействие (inter-process communication) – метод как один процесс может взаимодействовать с другим процессом. Сигналы нужны, чтобы асинхронно один процесс мог послать сигнал другому процессу. Процесс может либо обработать полученный сигнал, либо игнорировать полученный сигнал, либо выполнить действия по умолчанию (это определяется ОС).

Есть сигналы, которые нельзя игнорировать и нельзя обрабатывать.

Можно данные передать через файл – это долго и тратится ресурс диска/флешки.

Методы передачи:

почтовый ящик (относительно долго, но быстрее, чем через файл),

труба (псевдо-файл, у которого есть вход и выход (ФИФО), с одной стороны пишут, с другой читают в той же последовательности) – бывают именованные и неименованные (используются между родственниками),

разделяемая память (самый быстрый метод межпроцессорного взаимодействия, нет системных вызовов, следовательно, нет переключения),

сокеты (между процессами, которые находятся в одной сети).