

2 лекция , конспект

1. Мы говорили говорили говорили , итог все дела ,

Все программы исполняются OS API/OS Framework'ом (обращаясь к ядру ОС)

очевидные вещи

утилиты - системные проги с узким useg'em (копи , паст и всё такое)

Нет другого способа работать с ОС , кроме Frameworka OS

Никакой мистики

2. Системный вызов - спец механизм вызова прикладной программы ОС (которая не в ядре , но вызывают его функции) . СВ вызывается с помощью механизма ПРОГРАММНОГО ПРЕРЫВАНИЯ . Для этого процессор вызывает СВ своими инструкциями . для x86 - инструкция int , для x86_64 - SYSENTER/SYSEXIT , AMD - SYSCALL/ SYSRET . В составе ядра есть проги , которые являются обработчиками прерываний (handlers) . Прерывание - механизм , мы можем преостановить выполнение работы проца чтобы он исполнял другую прогу . После выполнения , он возвращается к моменту , где его прервали.

ПП - программное прерывание.

Мы как программисты не можем вызвать инструкцию int напрямую . Поэтому вызываем с помощью спец функцию ПП

процессы , потоки , диспетчеризация , отсчёт времени , управления память , i/o , синхронизация - топики

дальше скрины всякие обсуждает для проверки процессов , всё по лекции

Смелов рисует вилку онлайн без регистрации и комментариев