История ОС

Эвм первого поколения (Ос пока не существует)

1943 – Баллистические таблицы. Баллистические таблицы, расчет 4 года.

Джон Мокли и Джон Эккерт, разработали первый компьютер ЭНИАК (электронный дифференциальный анализатор, скорость одного расчета 5 мин, используются лампы)

Для перепрограммирования используется перемычки (чтобы перепрограммировать компьютер нужно было вытащить провод из одного отсека и вставить в другой) Данные находятся на перфокартах. Десятичные счисления.

Архитектура фон Нейнамана

1. Переход к двоичной системе счисления (есть сигнал - 1, нет сигнала – 0)
2. Вместо перфокарт используется память (память это пронумерованые ячейки, для обращения хранится свой адреса)
3. Программа – это последовательные команды

ЭВМ второго поколения (появляется первые языки программирования)

В 50x годах появились транзисторы, их плюсы по сравнению с лампами:

1. Высокая надежность
2. Транзисторы не выделяют тепло
3. Большая скорость переключения
4. Плотная компоновка деталей

Недостатки:

1. Проблемы с соединением
2. Дороговизна производства

Появились первые алгоритмические языки (COBAL, ALGOL)

Появление компилятора

Появления новых ролей в работе с ЭВМ:

1. Управления ЭВМ
2. Обслуживание ЭВМ
3. Программирование ЭВМ

ЭВМ 3-4 Го поколения

3 поколение

Проблемы:

Ответить на вопрос как разместить на полупроводнике все компоненты

Ответ:

Использование фотолитографию

Как соединить между собой элементы

Ответ: напыление алюминия

Плюсы:

1. Автоматическое производство следовательно, уменьшение цены
2. Улучшилось надежность
3. Ускорение ввода
4. Появление магнитного диска
5. Появление мультипрограммирование (пока одна программа простаивает, может работать другая)
6. Принцип разделения времени
7. Появление UNIX

4 поколение

Начало 70-80х

1. Появление виртуальной памяти (выполнение нужного участка программы
2. Появление концепции базы данных
3. Первый пк -altair 8800
4. Появление apple
5. Появление linux

1969 Деннис Ритчи и Кен Томпсон - Появление unix на B

1970 распространение unix и языка C

1978 ответвление BSD

1. сетевое взаимодействие
2. виртуальная динамичная память
3. стек протоколов ccpip
4. файловая система FFS

Идеология Unix

1. Программа решает одну задачу
2. Программы хорошо работают вместе
3. Поддержка текстовых потоков (запись символов в файл)

Linux

Создатель Линус Торальдс

1991 – разработка linux, разработка терминала для получения почты, разработка линукс которая запускается Bourne shell

ФУНКЦИИ ОС

1. Выделение программ
2. Выделение памяти, загрузка памяти исполняемой программы
3. Обработка системных вызовов
4. Ввод и вывод данных

Стандартизированный доступ к устройствам ввода и вывода

1. Обеспечение безопасности
2. Организация сетевого стека

Монолитное ядро

1. По сути одна программа
2. Одно адресное устройство
3. Linux
4. UNIX

Микроядерная архитектура:

1. Маленькое ядро
2. Базовое управление памятью
3. Управление процессорами времени

И т.д.

Совместимость ОС

1. Двоичная (компиляция кода без перекомпиляции)
2. На уровне исходных кодов(перекомпилирования кода)

Абстракция архитектуры

1. Пользовательские программы
2. Ядро
3. Ассемблер
4. Микрокод
5. Аппаратное обеспечение

В unix все является файлом

Типы файлов:

1. Символьные устройства
2. Блочные устройства
3. Каталоги
4. Ссылки
5. Именованные каналы
6. сокеты

Символьные устройства - вид файла **устройства** **в** **UNIX**/Linux-системах, обеспечивающий интерфейс к **устройству**, реальному или виртуальному, с возможностью посимвольного обмена информацией. В отличие от блочного **устройства** **символьное** **устройство**, как правило, не обладает возможностями произвольного доступа.

**Ссылка** в **Unix**-подобных системах — **это** своего рода указатель, напоминающий указатели в языках программирования, с той лишь разницей, что он указывает на файл или каталог.

**Сокет** домена **Unix** (англ. **Unix** domain socket, UDS) или IPC-**сокет** (**сокет** межпроцессного взаимодействия) — конечная точка обмена данными, подобная Интернет-**сокету**, но не использующая сетевого протокола для взаимодействия (обмена данными). Используется в операционных системах, поддерживающих стандарт POSIX, для межпроцессного взаимодействия. Корректным термином стандарта POSIX является POSIX Local IPC Sockets.