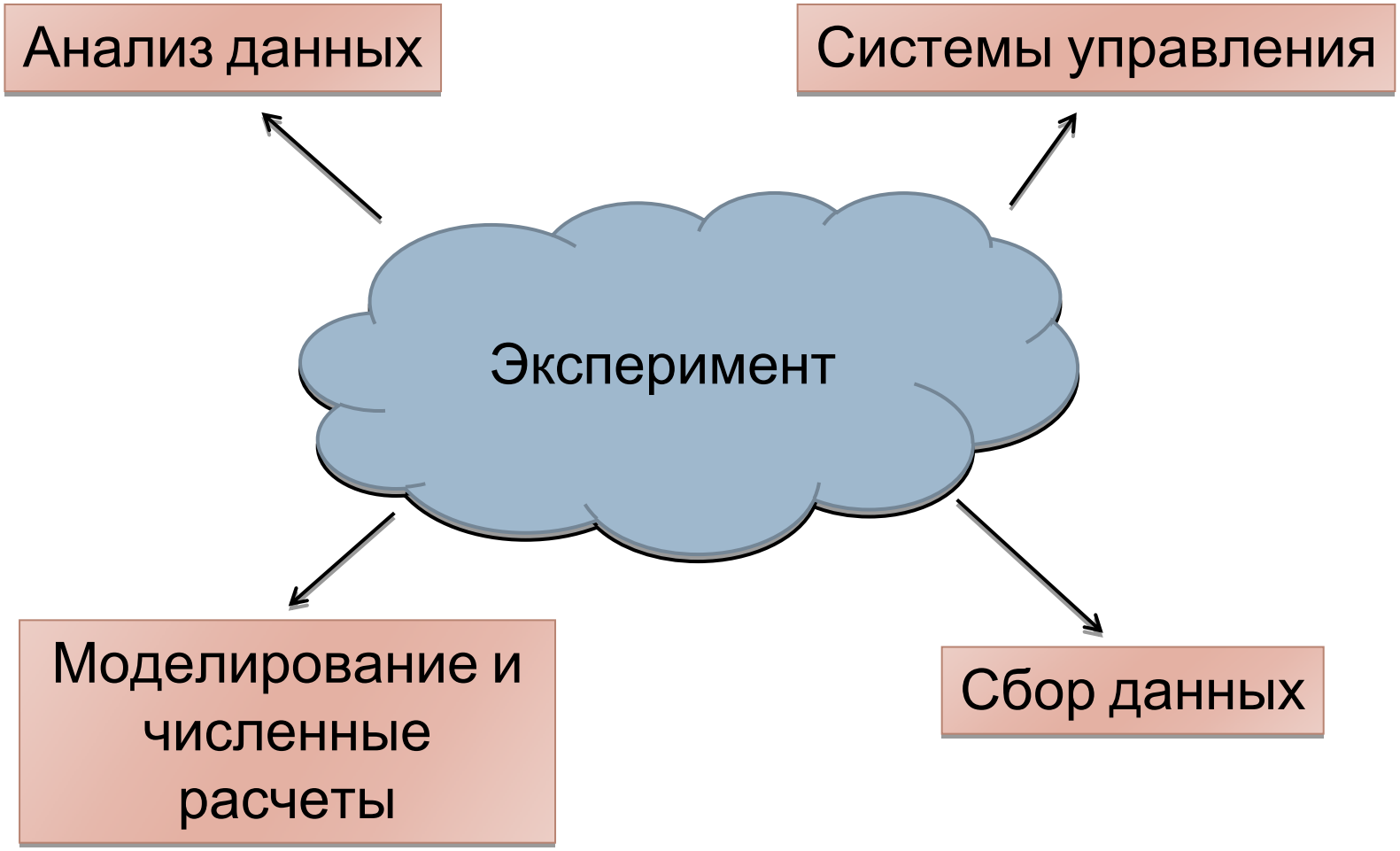
Компьютерные технологии в физике элементарных частиц.

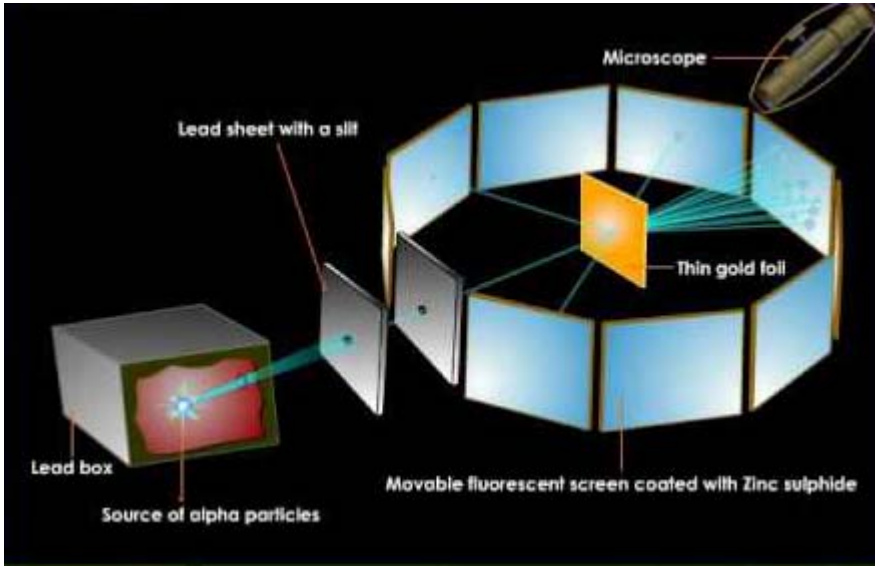
Введение. История вычислительной техники. Операционные системы.

Основные сферы применения компьютеров в физическом эксперименте

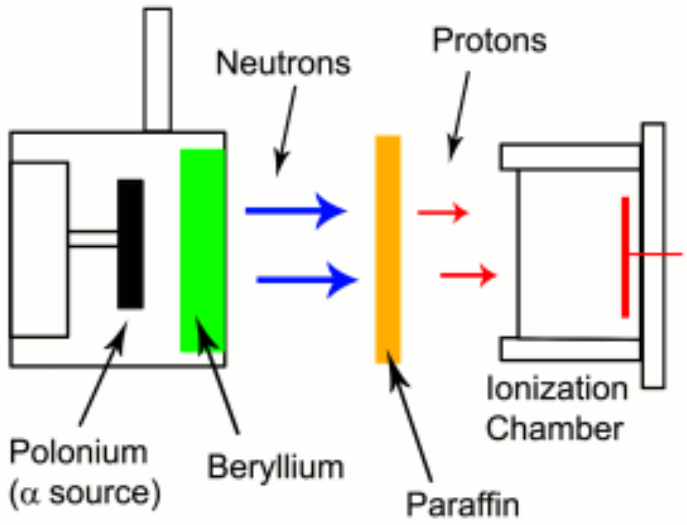


Примеры ранних экспериментов до появления компьютеров

В первой половине XX века физика развивалась без применения компьютеров.



Опыт Резерфорда (1911)

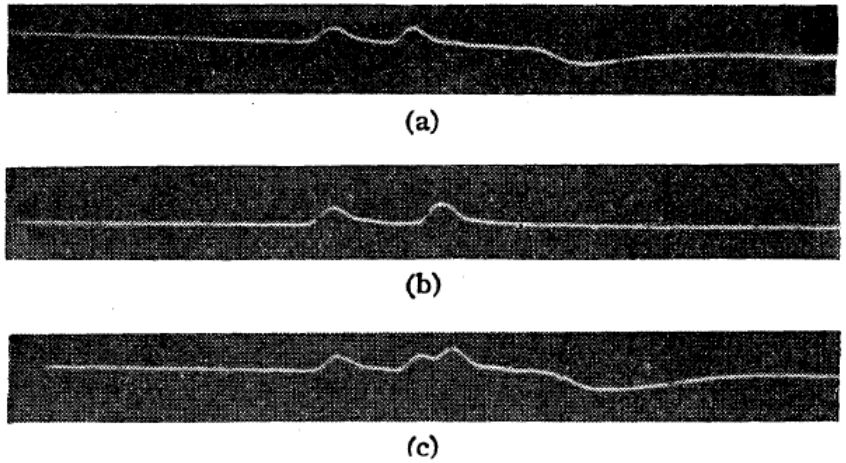


Открытие нейтрона (1932)

Даже Монте-Карло расчеты ядерного взрыва (1944) проводились без компьютеров – только с использование счетных машин (Р.Фейнман, “Вы, конечно, шутите, мистер Фейнман!”)

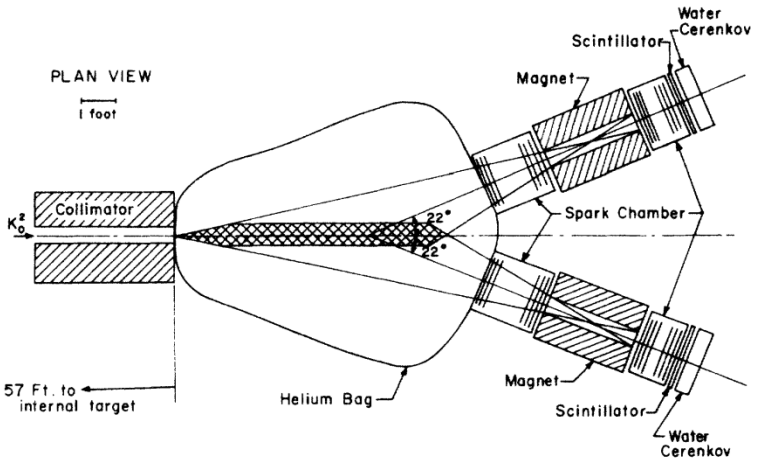
Примеры ранних экспериментов компьютерной эры

Начиная с 1950-х годов, компьютеры все чаще и шире использовались при проведении экспериментов.



Открытие антипротона (1955)

ССД: фотографирование дисплея осциллографа

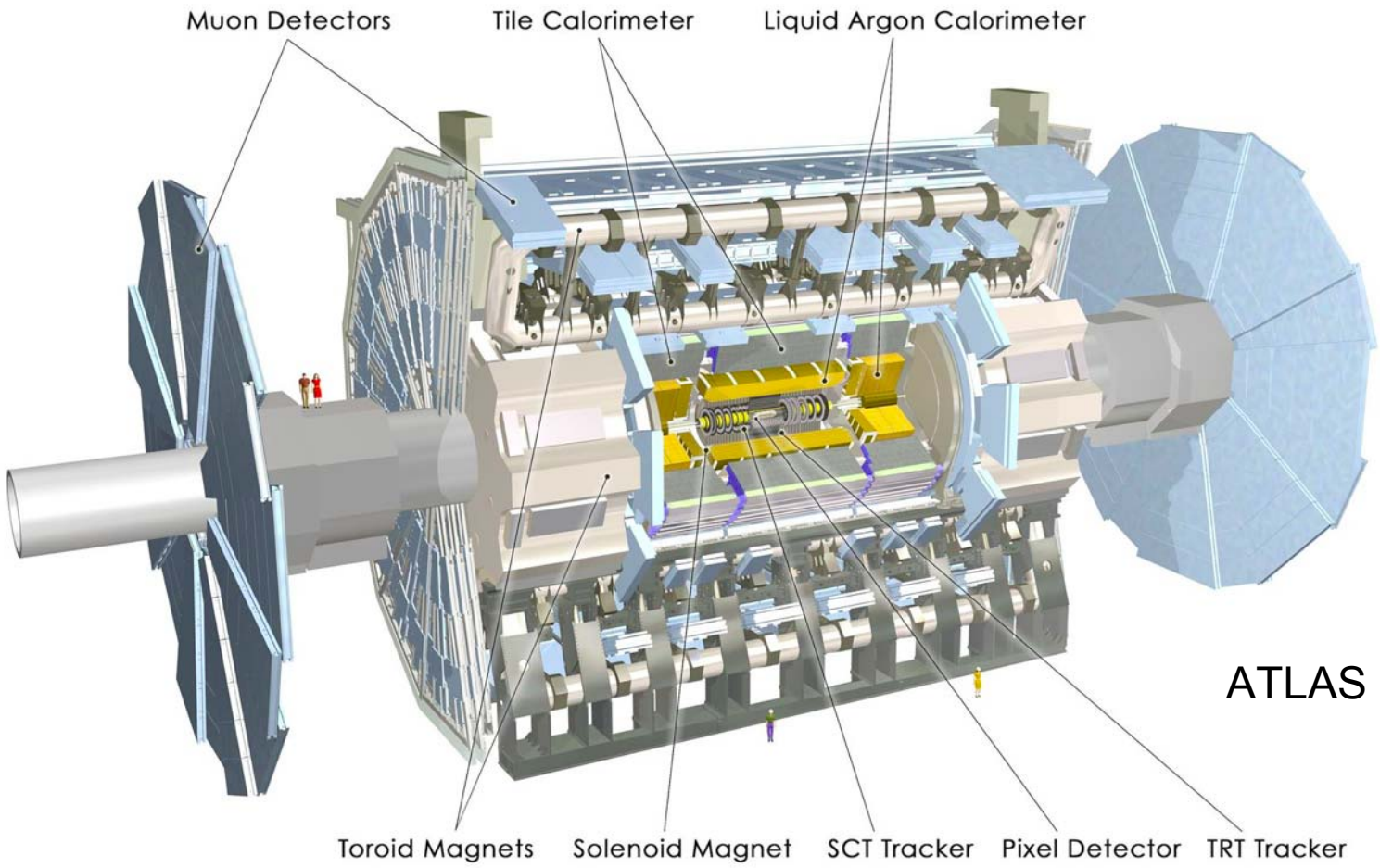


Открытие CP-нарушения (1964)

Компьютерная ССД и программа анализа данных

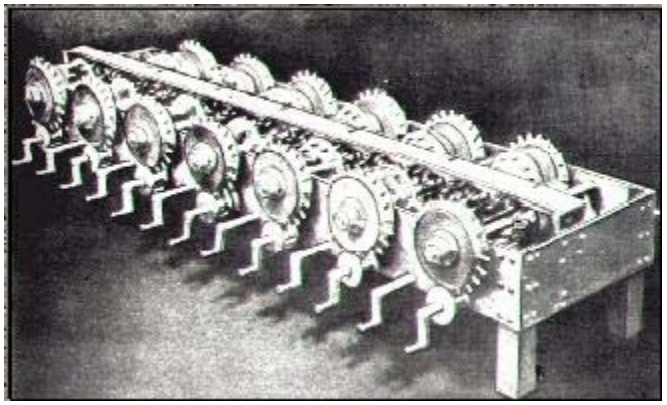
Современный эксперимент

Сегодня невозможно провести эксперимент без помощи компьютеров.

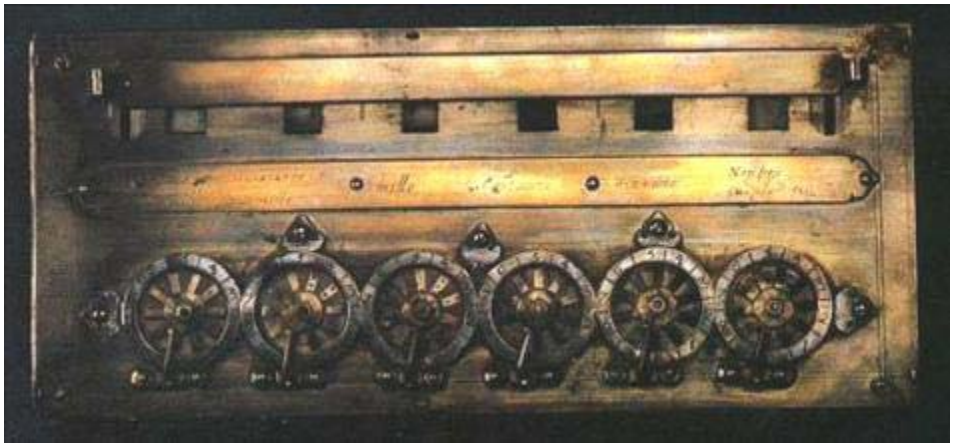


Механические компьютеры (1)

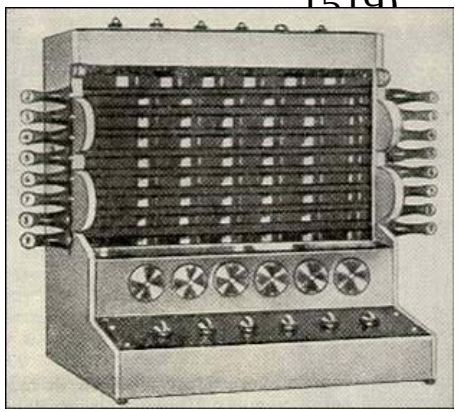
История вычислительных машин началась много веков назад.



13-разрядный сумматор Леонардо да Винчи (1452–1519)



“Паскалина” (1642)
Первая автоматическая
Сумматор
Автоматический перенос десятков, дополнительный код



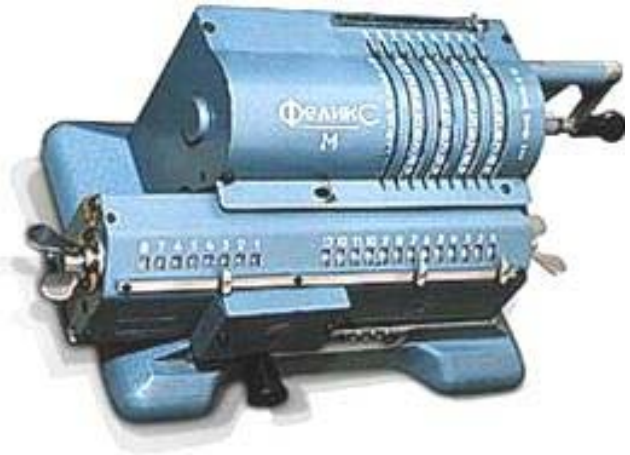
Машина Шиккарда (1623)
Полуавтоматическая
Сумматор, умножитель, хранение промежуточного результата
Использовалась И.Кеплером

Физики (астрономы ведь физики?) уже давно используют компьютеры!

Механические компьютеры (2)

Арифмометр Лейбница (1673)
12-разрядный автоматический сумматор, вычитатель, умножитель, делитель, вычисление квадратных корней

Арифмометр Томаса – серийно выпускался с 1818 до 1900



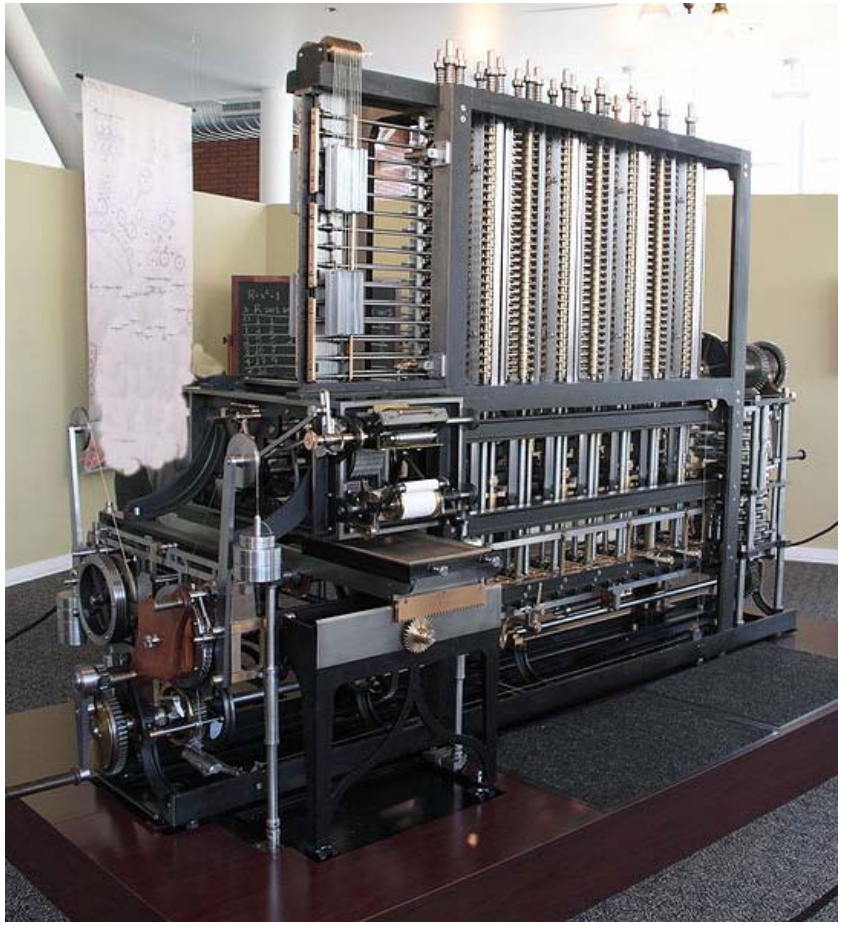
Арифмометр Однера (1874)
Выпускался до 70-х годов XX столетия

Разностная машина Бэббиджа

- 1797 – Г.Прони предложил схему организации вычислений:
  1. Построение математической схемы
  2. Определение последовательности операций (программирование)
  3. Выполнение программы

- 1802 – Ж.Жаккар изобрел перфокарты для программирования ткацкого станка

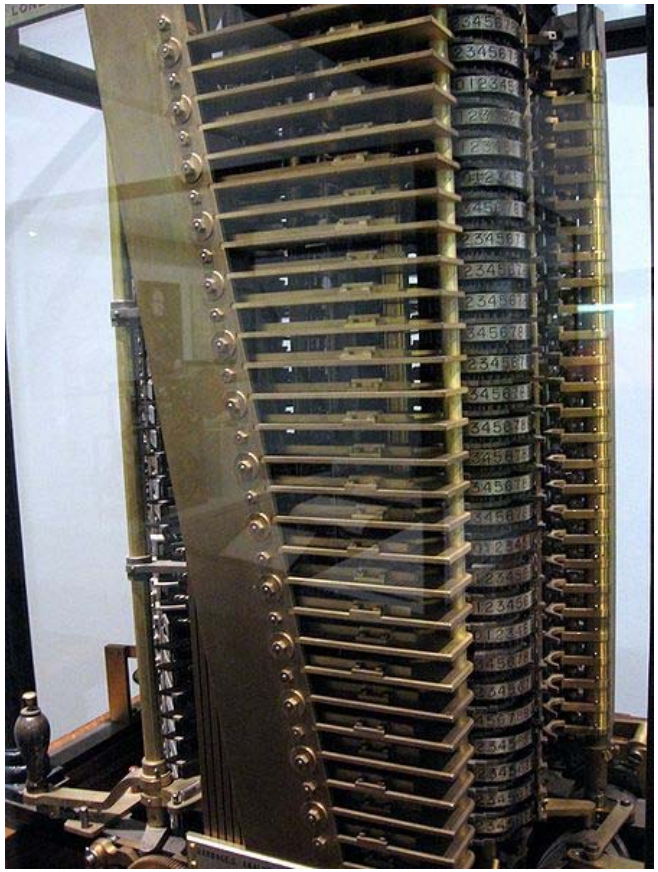
- 1820 – Разностная машина Ч.Бэббиджа, предназначенная для решения дифференциальных уравнений и табулирования многочленов
  - Жестко заданная программа
  - Полностью автоматическая
  - Наличие памяти
  - Наличие принтера



Аналитическая машина Бэббиджа

• 1840 – Аналитическая машина Ч.Бэббиджа

1. Программное управление
2. Разделение памяти (“склад”) и вычислителя (“мельница”)
3. Разделение команд и данных
4. а-



Часть машины из музея.

Машина никогда не была построена

Ада Лавлейс – первый программист

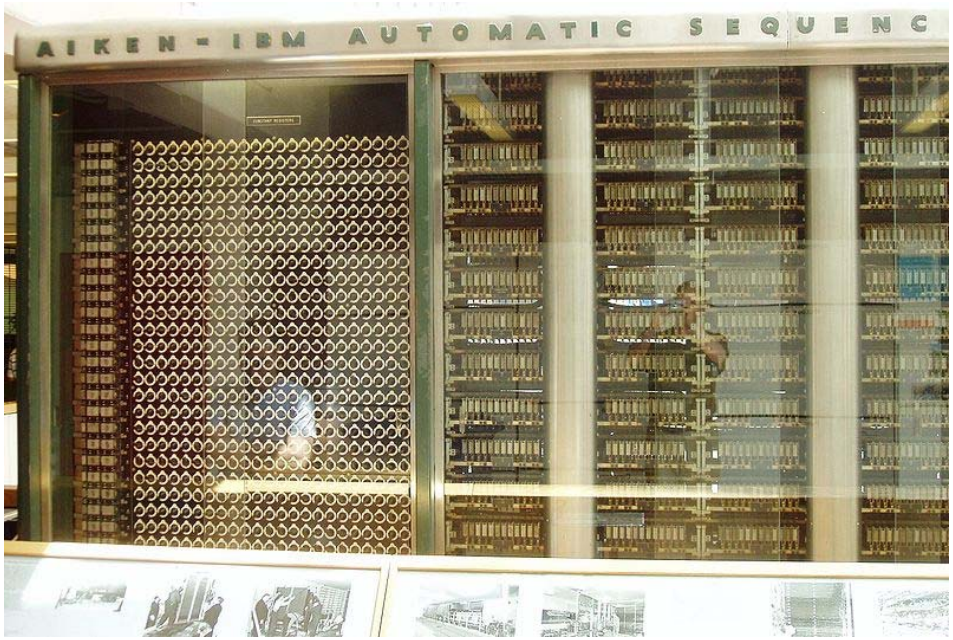
•предложила циклы, подпрограммы, условные переходы

•написала алгоритмы для аналитической машины

Электромеханические машины

В начале XX века появились электромеханические машины

- Счетно-перфорационные машины
  - Широко использовались в первой половине XX столетия в статистических исследованиях
- Машины К.Цузе (Германия)
  - Z-1 (1937), Z-2 (1939),…
- Машины Дж.Стибица (США)
  - Bell-1 (1939), Bell-2 (1942), …
- Машины Г.Эйкена
  - Mark-1 (1943)



- В этих машинах использовались как реле, так и механические элементы
- Вместо десятичной системы начали использовать двоичную систему
- Ввод информации через перфокарты или перфоленты
- Наличие относительно большой памяти

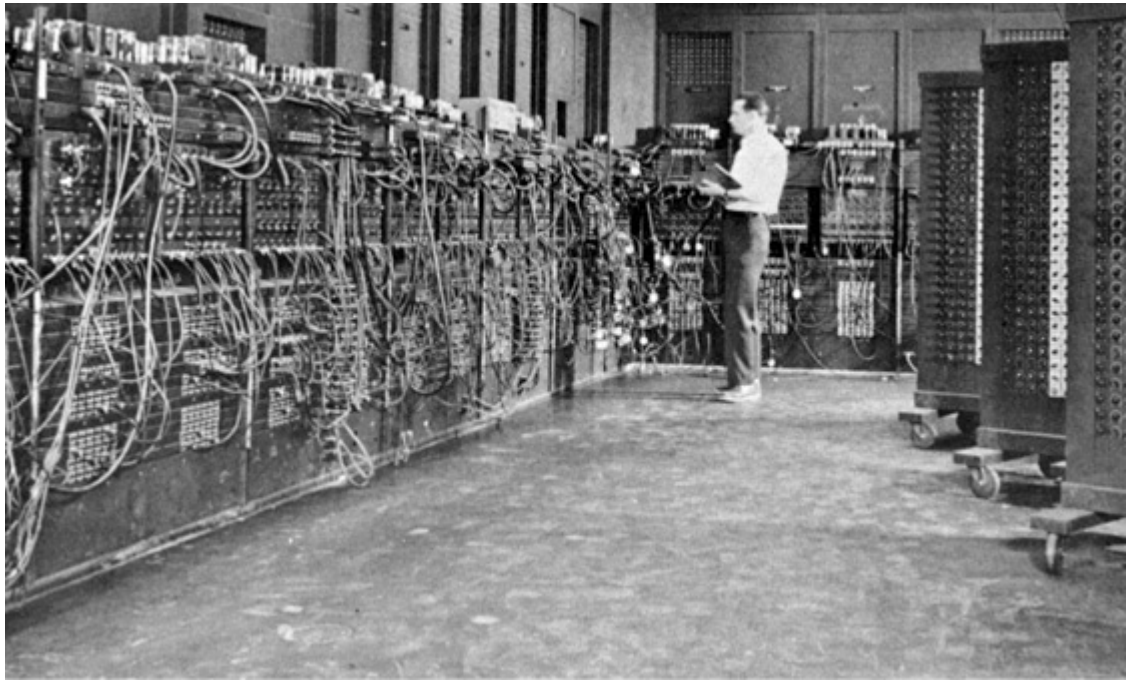
Первая универсальная электронная машина

14 февраля 1946 г. представлен ENIAC (Electronic Numerical Integrator and Calculator) – первая ламповая ЭВМ.

30 тонн, ~20000 ламп, память на 20 10-разрядных чисел

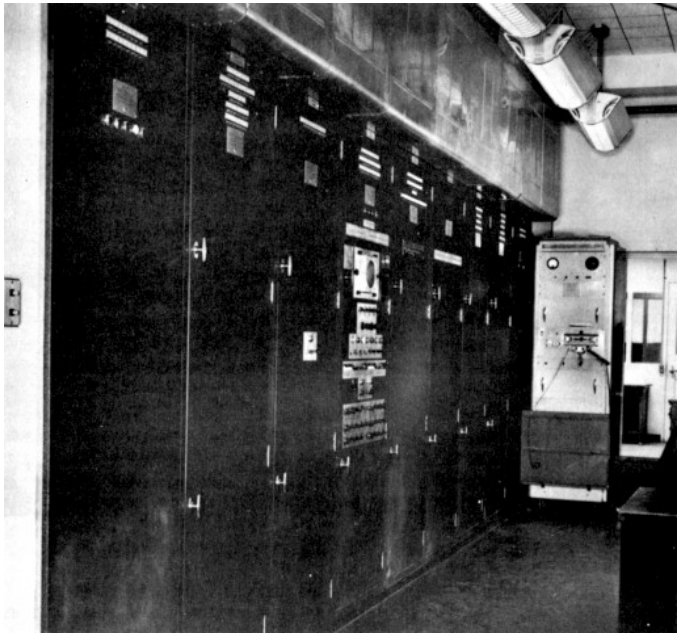
Быстродействие в 1000 раз больше механических машин!

Программирование с помощью перемычек.



Примеры первых компьютеров архитектуры фон Неймана

www.phys.nsu.ru



EDVAC (***E**lectronic **D**iscrete **V**ariable **A**utomatic **C**omputer*) (1951)
6000 ламп,
память 1000 44-битовых слов
Сложение 1200 Гц, умножение 400 Гц

UNIVAC (**UNIV**ersal **A**utomatic **C**omputer) (1951)
5000 ламп,
память 1000 12-разрядных чисел
Сложение ~10 кГц, умножение ~500 Гц
Первый коммерчески-успешный проект

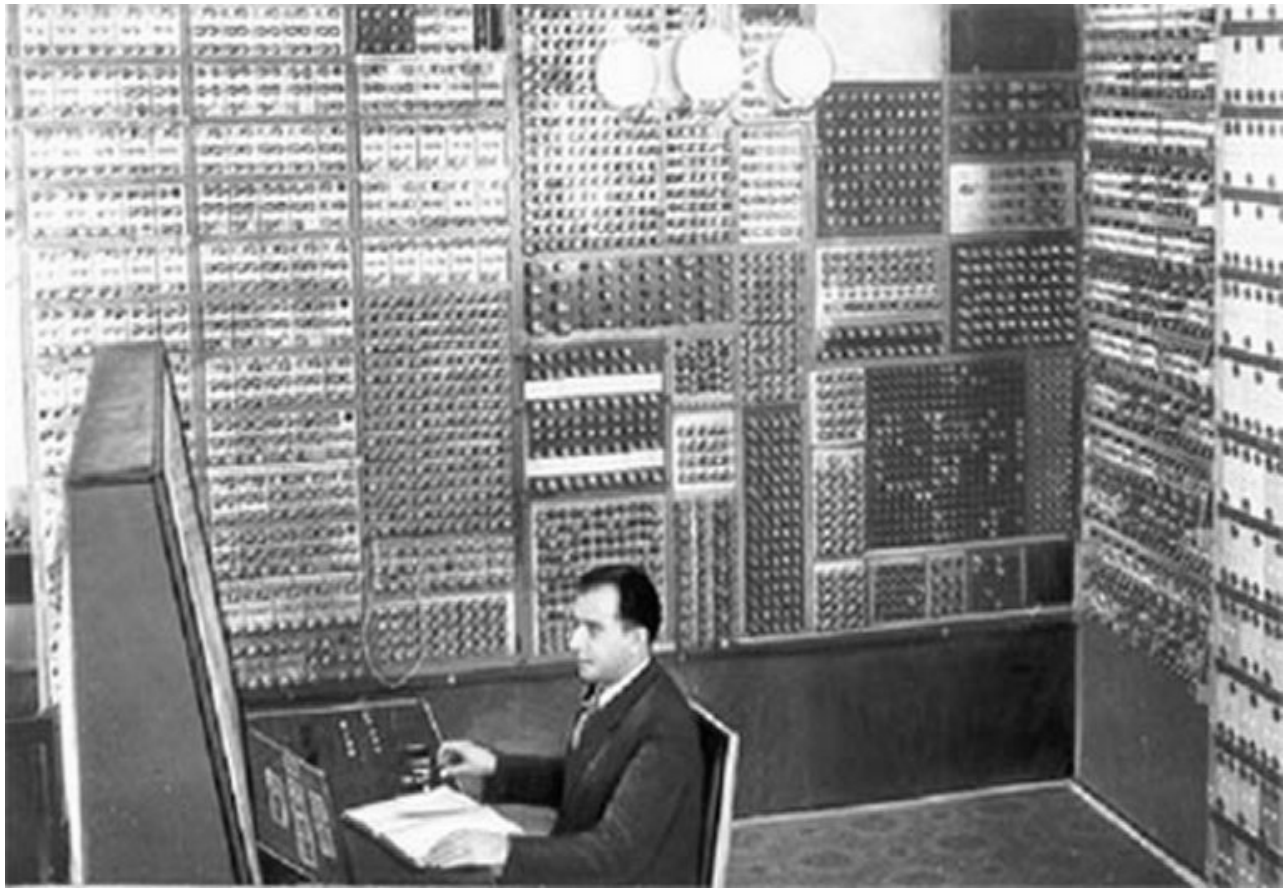
Первые зарубежные ЭВМ

Название	Год	Система счисления	Вычислительный механизм	Универ-сальный
Zuse Z3 (Германия)	1941	2	Эл.-механический	Да
Компьютер Атанасова—Берри (США)	1942	2	Электронный	Нет
Colossus Mark 1 (Великобритания)	1944	2	Электронный	Нет
Гарвардский Марк I — IBM ASCC (США)	1944	10	Эл.-механический	Нет
ENIAC (США)	1946	10	Электронный	Да
Манчестерская малая экспериментальная машина (Великобр.)	1948	2	Электронный	Да
EDSAC (Великобритания)	1949	2	Электронный	Да
CSIRAC (Австралия)	1949	2	Электронный	Да

Первая советская ЭВМ

МЭСМ (Малая электронно счетная машина) (1951)

- 6000 ламп
- ЗУ 96 16-битовых слов
- Быстродействие 50 Гц



Советские ЭВМ первого поколения

Название	Год выпуска	Элементная база	Быстро-действие	Разряд-ность	Объем ОП	Внешняя память
МЭСМ	1951	Электронные лампы	3000 оп/с	16	94 слова	Отсутствует
БЭСМ-1	1953	Электронные лампы	8 тыс. оп/с	39	2048 слов	На магнитных барабанах и магнитных лентах
М-3	1956	Электронные лампы	30 оп/с	30	2048 слов	На магнитных лентах
Урал-1	1957	Электронные лампы	100 оп/с	36	1024 слов	На магнитных лентах и перфоленте
М-20	1958	Электронные лампы и полупроводниковые	20 тыс оп/с	45	4096 слов	На магнитных барабанах и магнитных лентах

Второе поколение ЭВМ (1955-1965)

Замена электронных ламп на транзисторы резко увеличила способности ЭВМ.

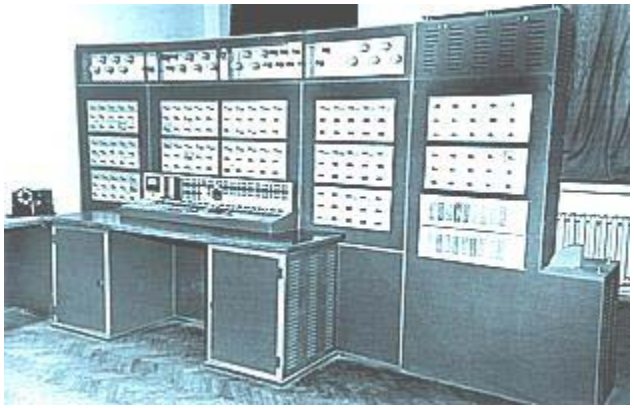
1. Производительность возросла с десятков кГц до сотен кГц
2. Объем оперативной памяти возрос в сотни раз, до сотен тысяч слов
3. Габариты, энергопотребление и надежность улучшились на порядок
4. Новые устройства ввода-вывода, первые магнитные диски
5. Стали появляться языки высокого уровня
6. Пакетная обработка

Зарубежные ЭВМ 2-го поколения

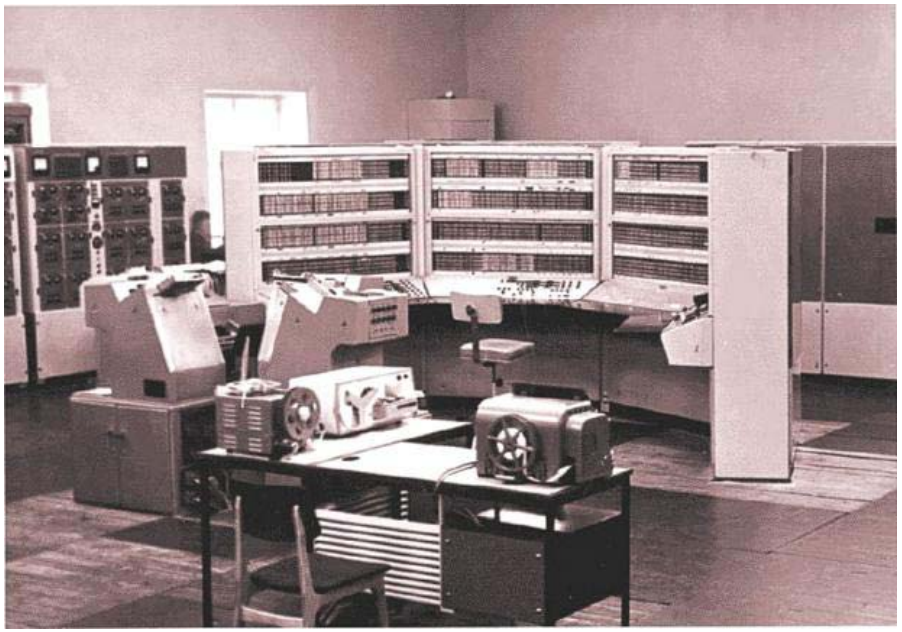
IBM 1401

PDP-1

Советские ЭВМ 2-го поколения

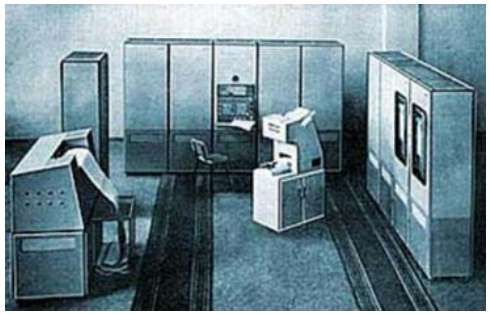


Сетунь (1959) - Троичная логика!



БЭСМ-6 (1966)
Первая суперЭВМ – 1 млн.оп./с
Конвейер, look-ahead,
виртуальная память
48-битовые слова, 10 МГц
Выпускалась до 1987

Минск-2 (1963)



Семейство ЭВМ Урал (1964)

Третье поколение ЭВМ (1965-1980)

Замена транзисторов на интегральные схемы привела к появлению качественно новых способностей ЭВМ:

1. Микропроцессоры
2. Многопроцессорные системы
3. Виртуальная память
4. Виртуальные машины
5. Разделение времени

Помимо этого, в этот период происходит быстрый рост производительности, объема оперативной памяти, надежности и т.п.

Периоды развития компьютеров второго и третьего поколения заметно перекрывались.

IBM System/360 – System/370

Семейство больших ЭВМ
Широко распространены в 70-х

Единый машинный код
Использовался микрокод
Виртуальная память
Виртуальные машины

Модель S/360	30	60
Частота, МГц	1	4
ОЗУ, кБайт	65	512
Производитель.		x21

PDP-11

Семейство миниЭВМ производства DEC
Широко распространены в 70-х, очень популярны в университетской среде

Ортогональная система команд
Единая шина (UNIBUS, Q-Bus)
Разработан для массового производства

16-битовые PDP-11 впоследствии были заменены 32-битовой VAX

«Рабочие лошадки» в большом количестве физических экспериментов

Советские аналоги

Семейство ЕС ЭВМ

Семейство СМ ЭВМ

Наша эра (1980-)

Что дальше?

1.Эра персональных компьютеров (1980-)

Разработка все более функциональных СБИС позволила создать компьютер на одной плате

2.Новые технологии периферийных устройств, систем хранения, памяти и т.п.

3.Развитие сетевых технологий

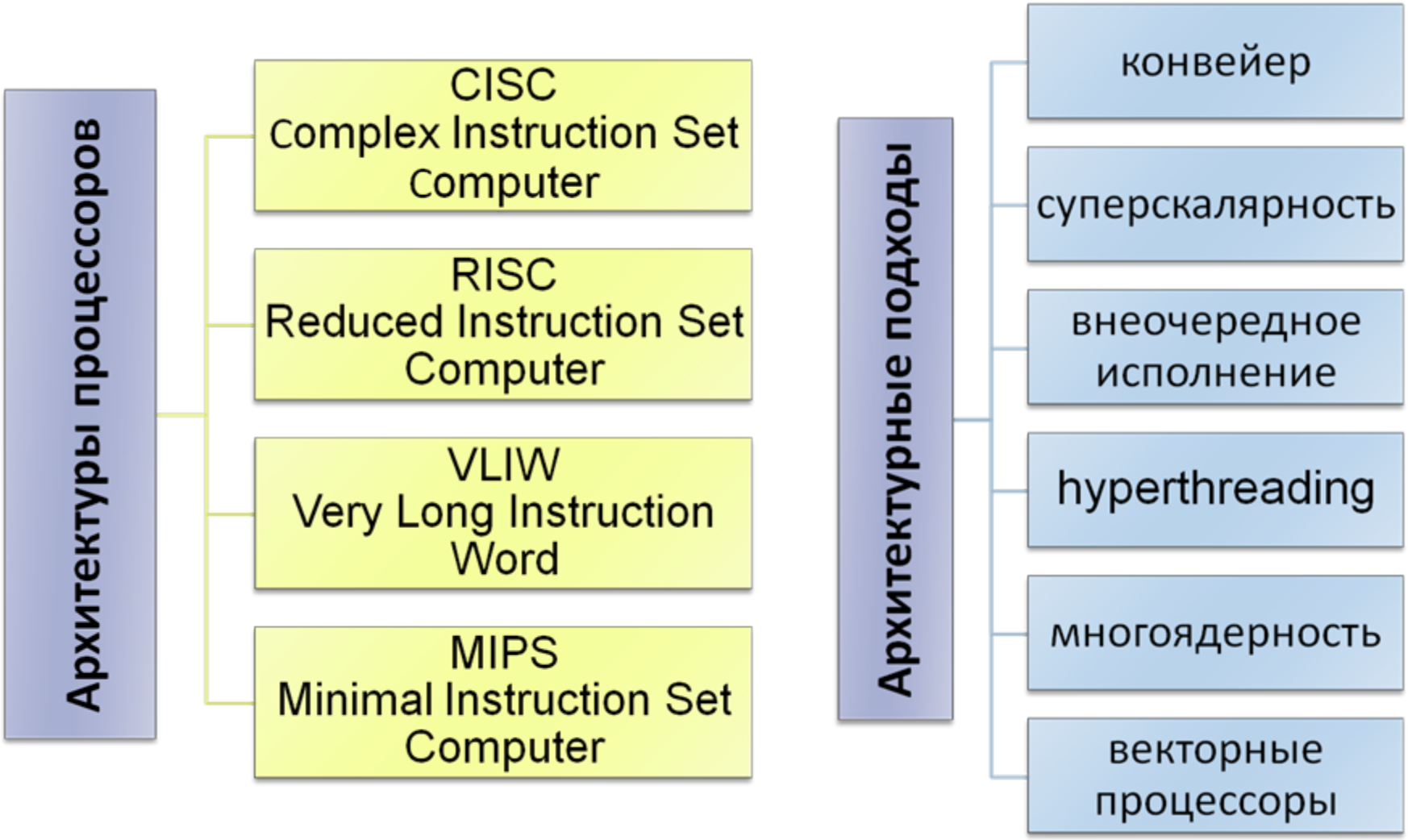
70-е – в лабораториях
80-е – в университетской и профессиональной среде
90-е – Интернет вышел в мир
Сейчас – повсеместное использование Интернета

4.Объединение многих процессоров

суперкомпьютерные фермы, ГРИД, «облака»

Современные типы компьютеров

Современные типы процессоров



Что такое операционная система?

Основные функции операционной системы:

- Абстрагирование пользовательских программ от деталей аппаратного обеспечения
- Выполнение привилегированных операций (режим ядра) – системные вызовы
- Менеджер ресурсов

Типы операционных систем

По сфере применения:

- ОС мэйнфрэймов (z/OS)
- Серверные (Linux, Window Server 2003)
- Персональные (Linux, Windows XP)
- Мобильные (Palm OS, iOS, Android)
- Встроенные (VxWorks, QNX)

По свойствам:

- Однозадачные (MS-DOS)
- Многозадачные (Linux)
- Многопроцессорные (Linux)
- Реального времени (eCos)

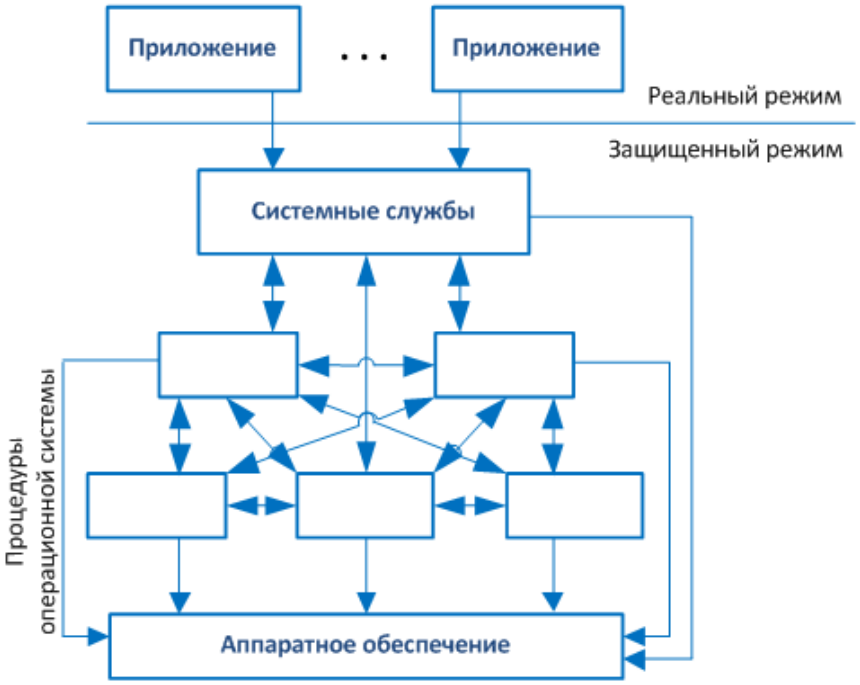
По количеству пользователей:

- Однопользовательские (MS-DOS)
- Многопользовательские

По интерфейсу:

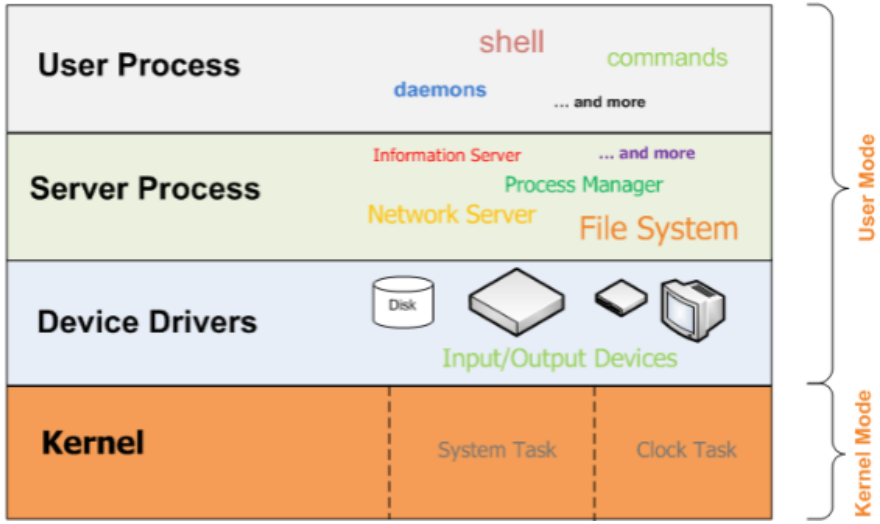
- Консольные
- Графические

Организация кода операционных систем



Монолитные системы

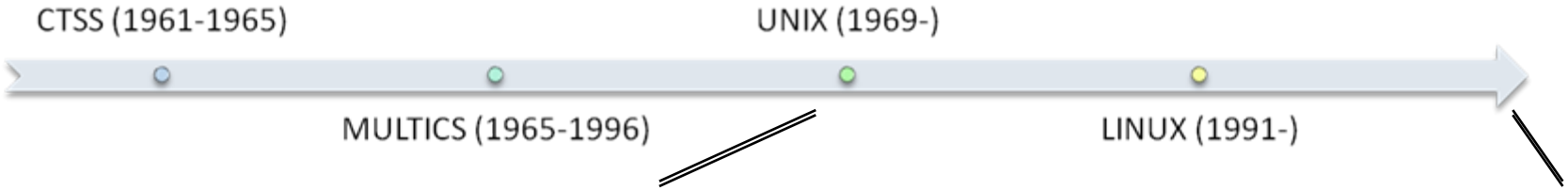
Весь код ОС работает в режиме ядра, в общем адресном пространстве. Возможна модульная организация.



Микроядро

Только небольшая часть ОС (основные функции) работает в защищенном режиме.

UNIX



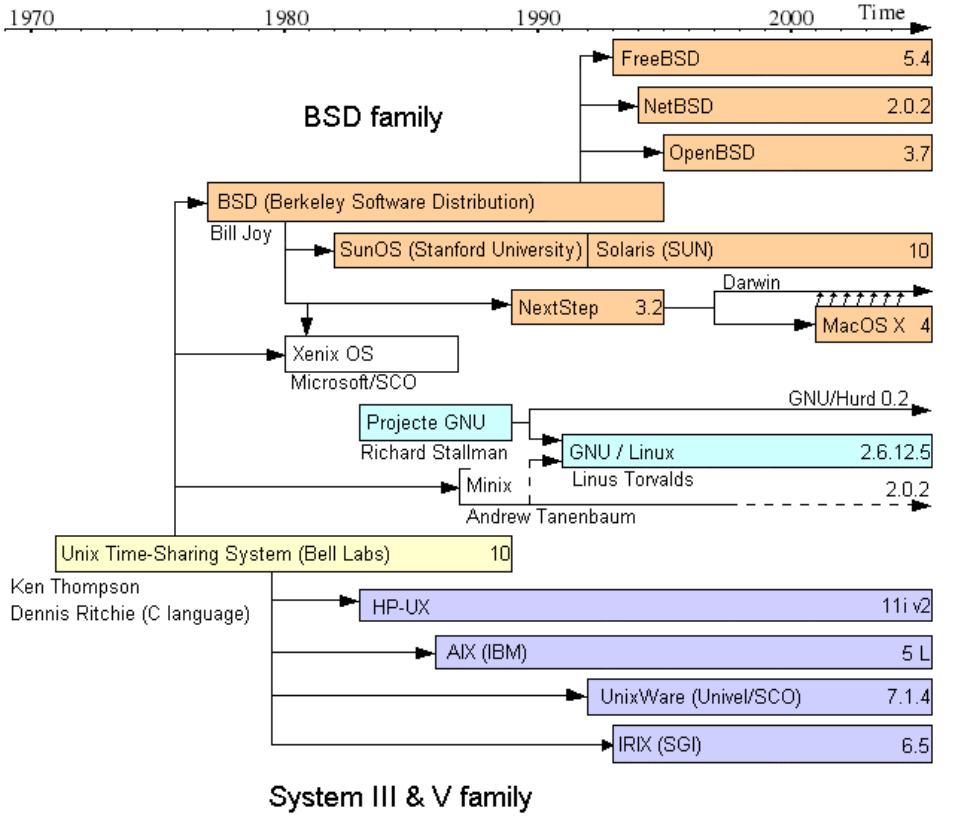
1964-1969 – ОС MULTICS (**Mult**iplexed **I**nformation and **C**omputing **S**ervice)

1969 – Первая версия UNIX (UNIplexed Information and Computing System) – К.Томпсон, Б.Керниган

1975 – Версия, написанная на C

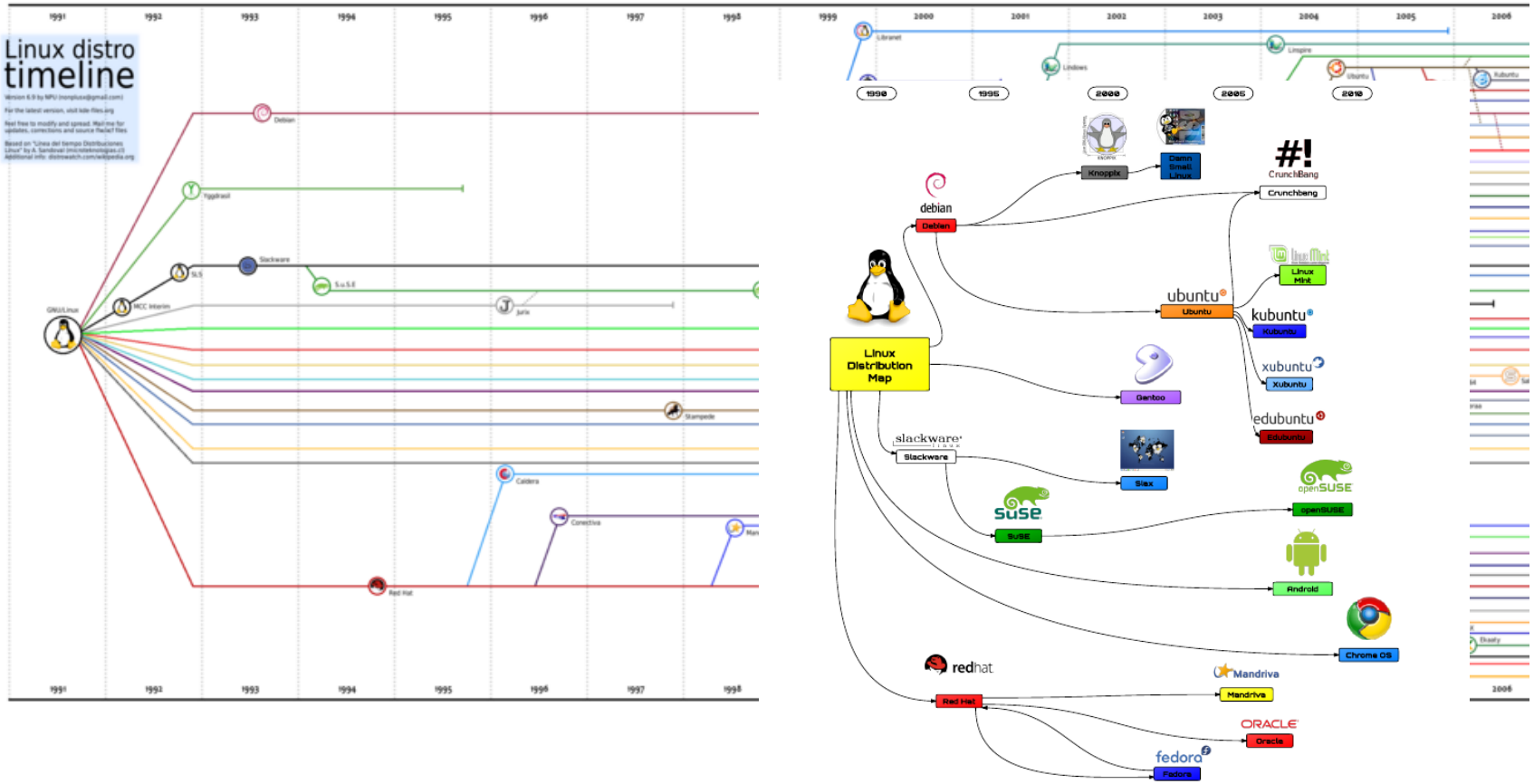
Начало 80-х – разделение на BSD и System V

1991 – появление GNU Linux



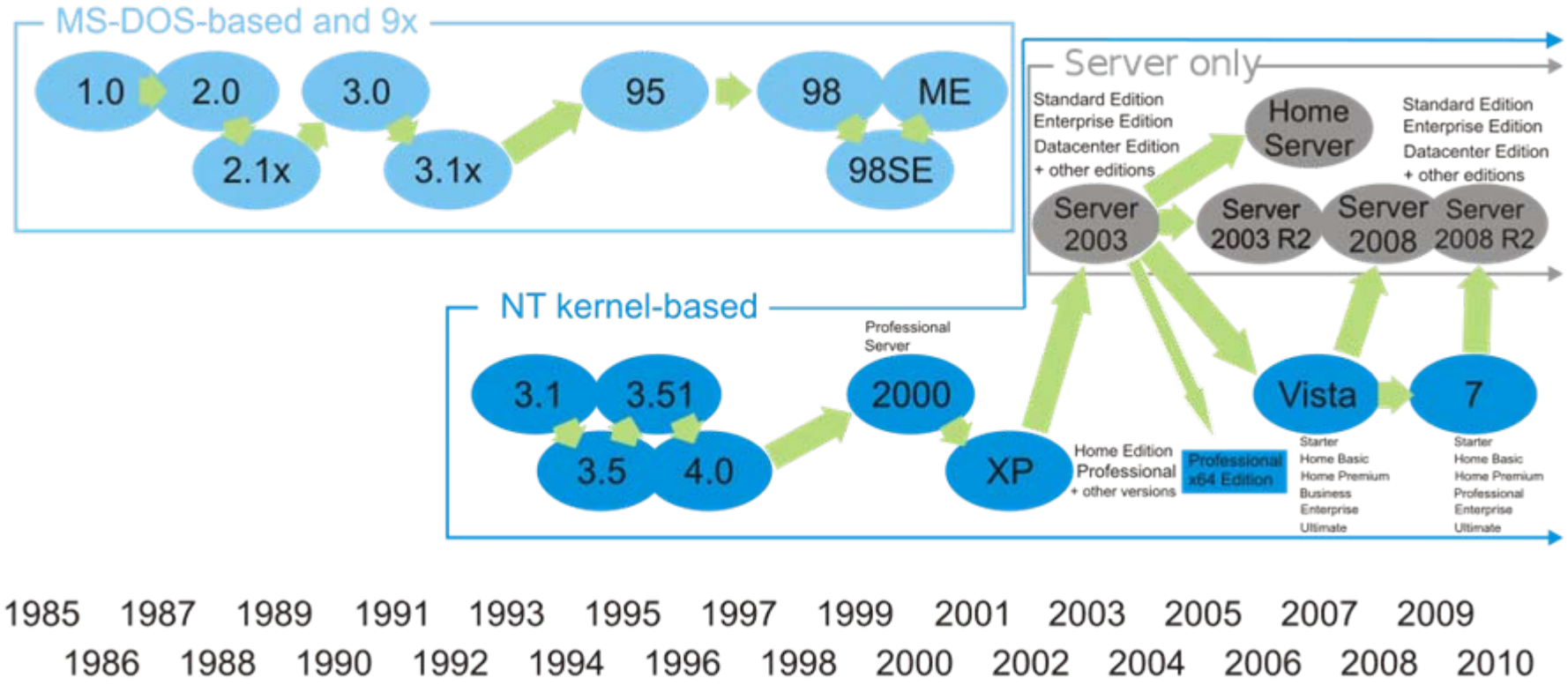
LINUX

LINUX появился как open-source реализация UNIX и быстро стал популярен благодаря партнерству с GNU. Автор - Linus Torvalds.



Windows

Windows разрабатывался как графический интерфейс (Apple-like) для тогда новых IBM PC и получил широкое распространение благодаря буму персональных компьютеров.



Основные абстракции ОС

Одна из основных задач операционной системы – предоставление пользова-тельским программам удобных абстракций, скрывающих сложные детали аппаратного обеспечения.

Ключевые абстракции операционных систем:

Процессы

В современных системах центральный процессор одновременно (почти) обрабатывает много заданий, постоянно переключаясь между ними. Эти задания называются процессами.

Процессор обрабатывает один процесс в конкретный момент времени (на 1 ядре). При переключении в памяти сохраняется контекст процесса.

В UNIX образуется иерархия процессов – родительские процессы порождают дочерние. Адресное пространство дочерних процессов отличается от родительских.

Порождение процессов:

UNIX : int fork(void)
Windows: CreateProcess(…)

Замена процесса:

UNIX : exec()

Контекст процесса

Пользовательский уровень	Регистровый уровень	Системный уровень
• Текст • Данные • Стек • …	• Счетчик команд • Указатель стека • Регистры • …	• Запись в таблице процессов • Адресные таблицы • …

Потоки

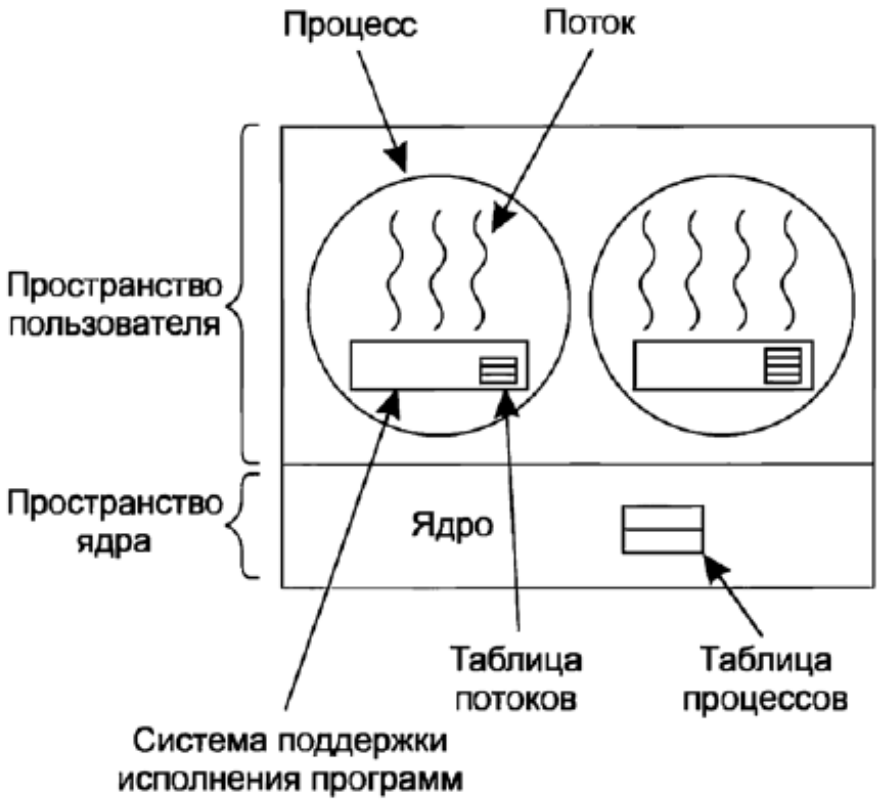
Внутри отдельного процесса можно создать множество потоков. Переключение потоков не требует переключения контекста процесса. Потоки выполняются в едином адресном пространстве (хотя у каждого есть свой стек).

Типичные многопотоковые программы: интерактивные программы, сервера, реагирующие на запросы

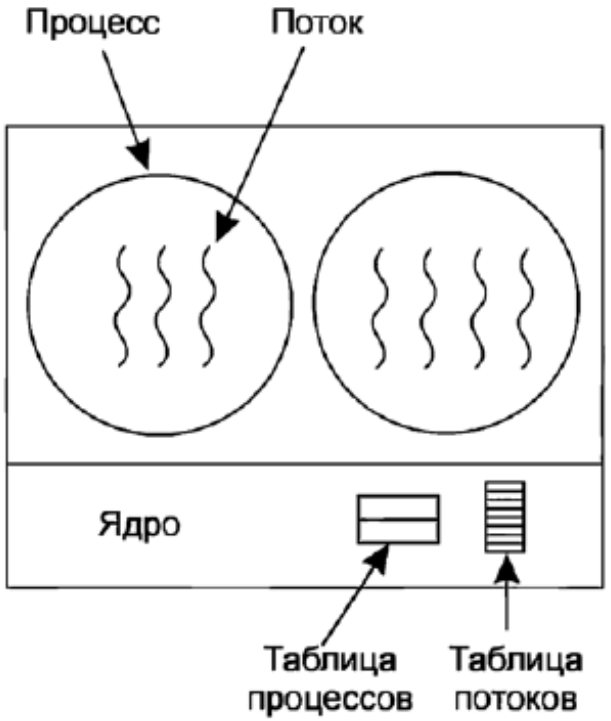
Элементы, общие для всех потоков одного процесса	**Элементы, присущие каждому потоку**
Адресное пространство	Счетчик команд
Глобальные переменные	Регистры
Открытые файлы	Стек
Дочерние процессы	Состояние
Сигналы и обработчики	
Учетная информация	

Управление потоками

Управление потоками (переключение) возможно как на пользовательском уровне, так и на уровне ядра.



Потоки на пользовательском уровне



Потоки на уровне ядра (LINUX)

Планирование процессов

Одна из важнейших функций операционной системы – планирование процессов.

Планирование процессов { Когда приостановить процесс? / Какой процесс активизировать вместо него?

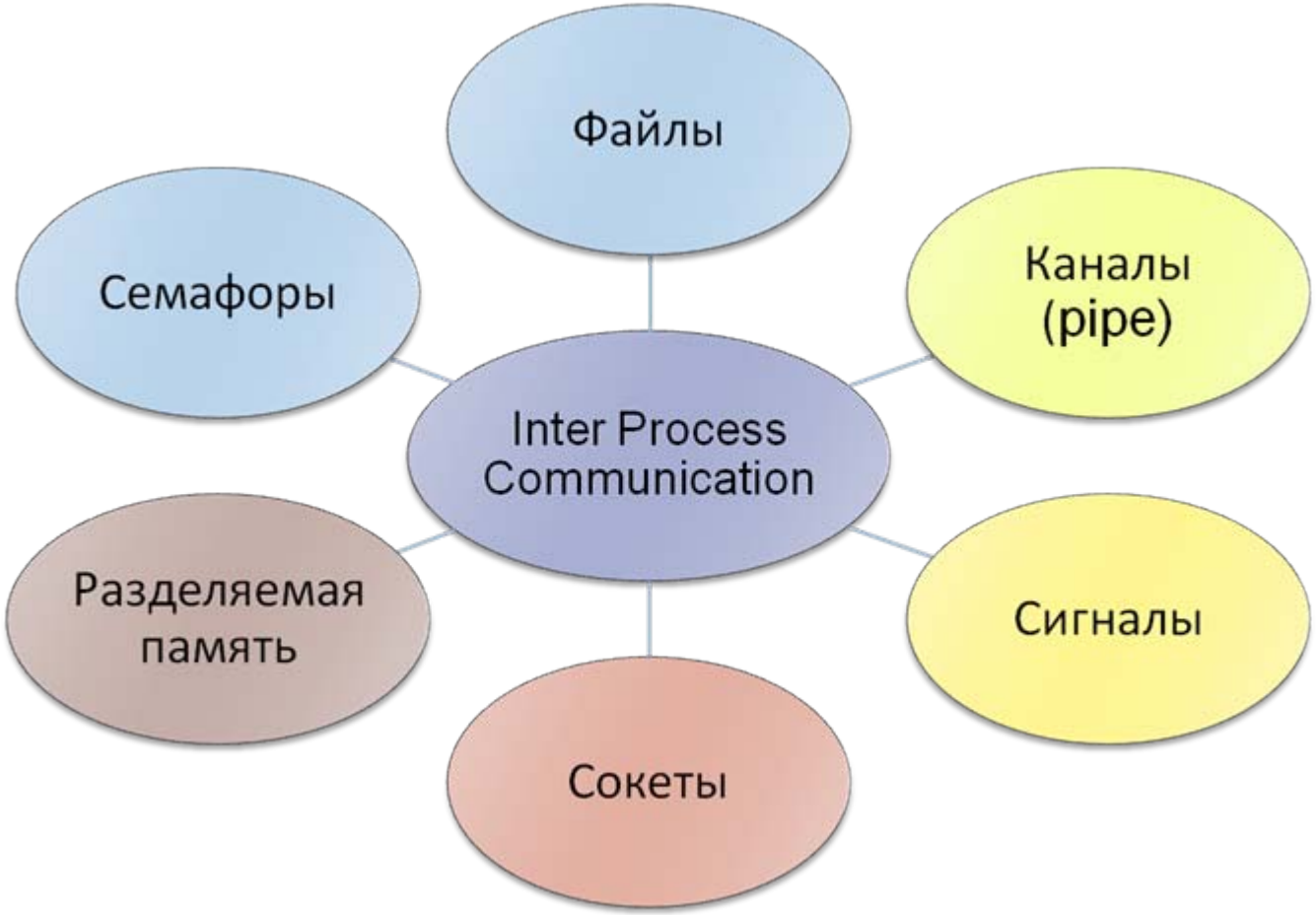
Два основных подхода: неприоритетное (процесс работает пока не освободит процессор) и приоритетное планирование (процесс работает выделенный квант времени).

В зависимости от типа системы применяются разные алгоритмы планирования.

- **Планирование в пакетных системах**: по мере поступление, сначала короткие задания,…
- **Планирование в интерактивных системах**: циклическое планирование, планирование с приоритетами, планирование с несколькими очередями, лотерейное планирование, справедливое планирование,…
- **Планирование в системах реального времени** (жесткого или мягкого): статическое планирование, динамическое планирование,…

Взаимодействие процессов

Операционная система предоставляет множество механизмов для взаимодействия отдельных процессов.



Управление памятью

Другой важнейшей задачей операционной системы является управление памятью.

Первые программы напрямую работали с физической памятью. Основной недостаток – невозможно одновременно запустить несколько программ. Такое решение до сих пор используется во встроенных системах. Простое решение проблемы – добавить регистр начала области памяти.

Абстракция памяти – у каждого процесса есть свое независимое адресное пространство. Диспетчер памяти связывает адресное пространство процесса и физическую память.

Простое решение для отображения – использование базового регистра и своппинг.

Наиболее мощное и распространенное решение – использование виртуальной памяти. В этом решении память процесса может превышать объем физической памяти. Процессор передает диспетчеру памяти виртуальный адрес адресного пространства, а диспетчер памяти преобразует его в физический адрес.

Страничная организация памяти

Виртуальная память организована как множество страниц. Размер виртуальной памяти может превышать объем физической памяти. Запуск и исполнение программы возможно, даже если не все страницы находятся в памяти. Типичный размер страницы – 4 кБ.

Страницы подгружаются с диска независимо. Порядок страничных блоков не совпадает с порядком виртуальных страниц.

В физической памяти одновременно находятся страницы многих процессов. Для преобразования виртуальных адресов в физические используется таблица страниц процесса.

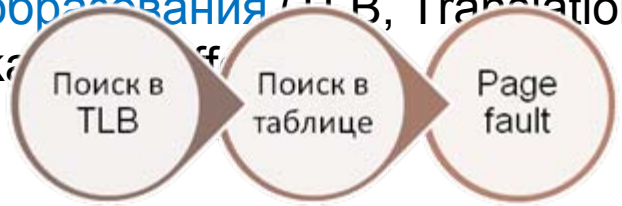
В некоторых системах у процесса есть несколько независимых адресных пространств – сегментов памяти. Например, в UNIX у процесса три сегмента: текстовый, сегмент данных и стек.

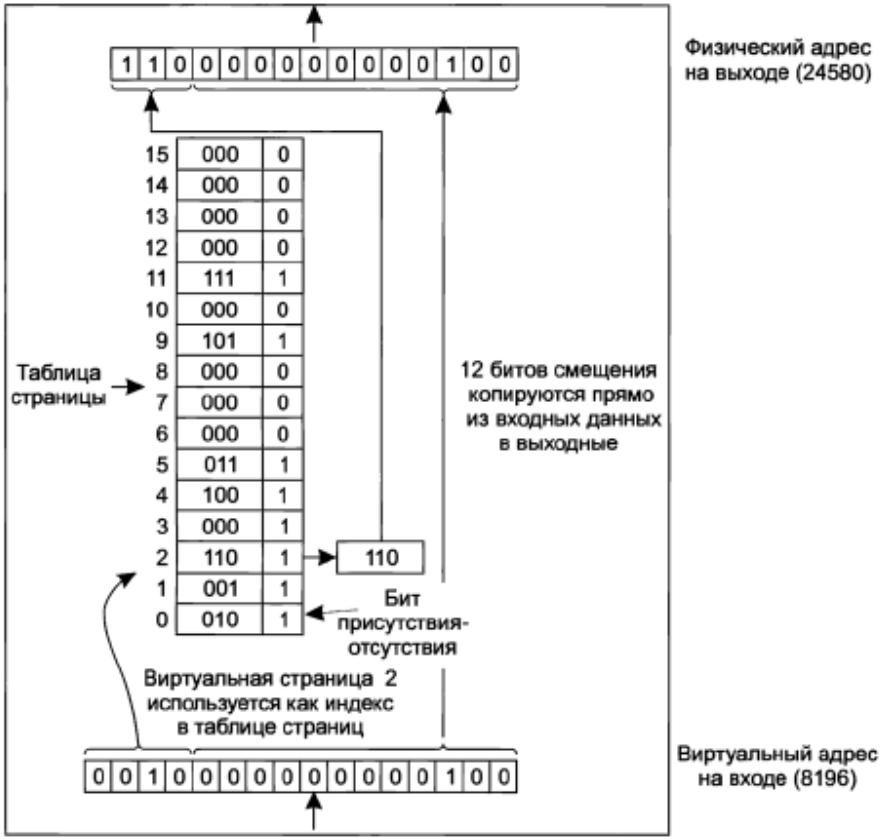
Преобразование виртуальных адресов в физические

Типичная запись в таблице страниц:

Для 32-битового адреса и страниц размером 4 кБ размер таблицы страниц 2^{20} записей, или несколько МБ. *Что делать для 64-битовых адресов?*

Для ускорения преобразования в процессоре присутствует ассоциативный буфер быстрого преобразования (TLB, Translation Look





Алгоритм преобразования виртуальных адресов в физические

Отдельные вопросы управления памятью

Управление памятью требует от операционной системы решения большого круга вопросов.

Многоуровневые таблицы страниц

- Позволяет уменьшить размер страницы и ускорить поиск

Алгоритмы замещения страниц в памяти

- Когда вся физическая память занята, какую страницу оптимальнее всего выгрузить?

Алгоритмы хранения страниц на диске

- Как организовать хранение выгруженных страниц на диске?

Совместное использование страницы несколькими процессами

- Используется при межпроцессном взаимодействии
- Динамические библиотеки
- Memory-mapped files

Файловая система

Файловая система – абстракция, которую предоставляет операционная система для доступа к долговременному хранилищу информации.

Основные требования:

- возможность хранения огромных объемов;
- срок хранения данных не связан с временем жизни процесса;
- возможность доступа к информации многих процессов.

Основные понятия – файлы и каталоги.

Файлы – основная единица хранения в файловой системе. Идентифицируется именем (в некоторых ОС и расширением), обладает набором атрибутов. С точки зрения операционной системы, файл – это последовательный набор байтов. Файлы бывают с последовательным или произвольным доступом.

Каталоги – специальные файлы, предназначенные для организации структуры файловой системы.

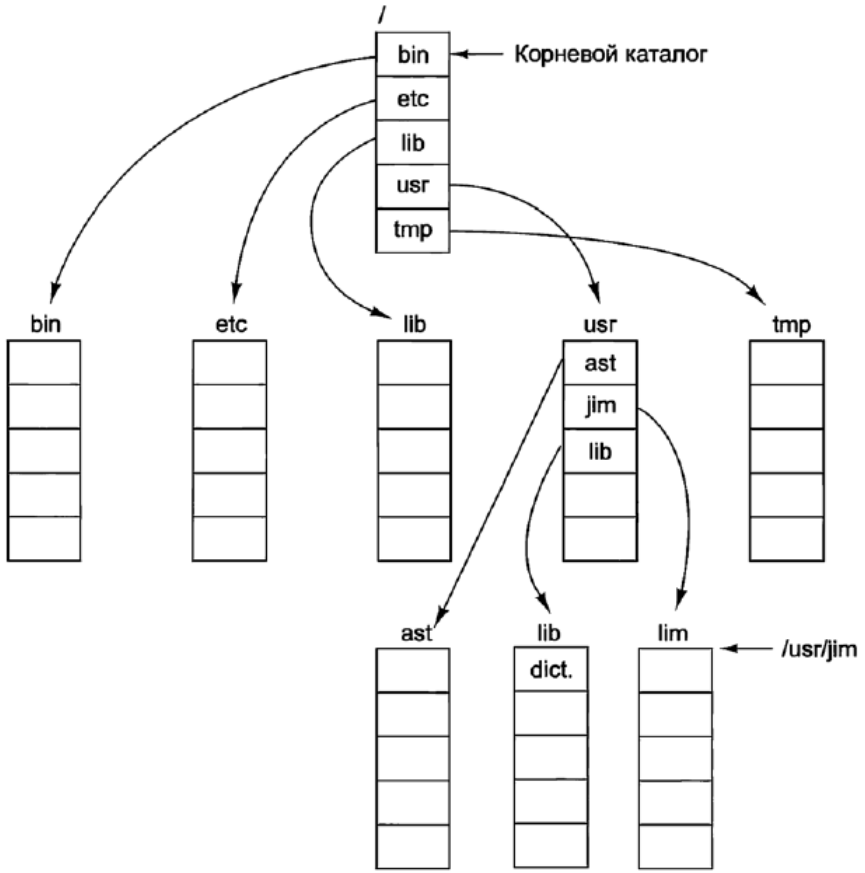
Также существуют специальные файлы – устройства ввода-вывода, информация о системе и т.п.

Каталоги

Каталоги образуют иерархическую структуру. В каталоге могут содержаться файлы или другие каталоги.

В UNIX есть корневой каталог.

В каталоге могут присутствовать символические линки (жесткие или мягкие) – ссылки на файл, который содержится в другом каталоге.



Файловая система ext2 (LINUX)