

Севастопольский государственный университет  
Кафедра информационных систем

Курс лекций по дисциплине  
«Алгоритмизация и программирование»

Лектор: Бондарев Владимир Николаевич  
Сметанина Татьяна Ивановна

# Лекция 1

Цель дисциплины.

История развития ЭВМ.

Понятие архитектуры компьютера.

Принципы фон Неймана.

## Цель дисциплины **АиП**

Общий объем дисциплины – **504 часа (14 кредитов ECTS)**

Форма обучения	Курс	Семестр	Лекции (час.)	Практические занятия (час.)	Лабораторные занятия (час.)	Аудиторные занятия (час.)	Самостоятельная работа (час.)	Общий объем (час.)	РГЗ	Курс. проект	Зачет (сем.)	Экзамен (сем.)
Дневная	1	1	36	18	36	90	126	216	РГЗ	-	-	1
	1	2	36	-	54	90	90	180		-	-	2
	2	3	-	18	-	18	54	72		КП	3	

**Целью дисциплины “Алгоритмизация и программирование” (АиП)** является:

- 1) обучение студентов типовым алгоритмическим структурам;
- 2) основным понятиям и управляющим конструкциям алгоритмических языков программирования (C/C++, Java);
- 3) основным структурам данных и алгоритмам их обработки;
- 4) формирование практических навыков разработки программ с помощью систем программирования (Dev-C++, Eclipse CDT/ MinGW).

# Содержание дисциплины

## "Алгоритмизация и программирование"

- архитектура компьютеров, принципы фон Неймана;
- позиционные системы счисления и представление данных в памяти компьютеров;
- понятие алгоритма и типовые алгоритмические структуры программирования;
- элементы алгоритмических языков: концепция типов данных, имена, значения, указатели, переменные, константы, операции, выражения;
- структурное программирование: следование, ветвление, циклы;
- процедурно-ориентированное программирование, рекурсия;
- методология разработки программ: нисходящее и восходящее проектирование, модульное программирование;
- организация данных (массивы, строки, структуры) и алгоритмы их обработки;
- файловые структуры данных;
- динамические структуры данных (списки, очереди, стеки, бинарные деревья) и алгоритмы их обработки;

# Лабораторные занятия

№	Наименование лабораторного занятия	Дневная ФО	
		Час.	Семестр
1	2	3	4
	<b>Основы работы на ПЭВМ</b>		
Лз-1	Основы работы в операционной системе Windows	4	1
	<b>Язык программирования C/C++</b>		
Лз-2	Программирование линейных и разветвляющихся алгоритмов	4	1
Лз-3	Программирование алгоритмов циклической структуры	4	1
Лз-4	Программирование алгоритмов обработки одномерных статических массивов	4	1
Лз-5	Обработка одномерных динамических массивов с помощью функций	4	1
Лз-6	Обработка двумерных массивов с помощью функций	4	1
Лз-7	Программирование с использованием модулей	4	1
Лз-8	Программирование операций над строками	4	1
Лз-9	Программирование операций над структурами и над массивами структур	4	1

# Основная литература

№	Наименование и полное библиографическое описание	Количество экземпляров в библиотеке
<b>Основная литература</b>		
1	Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона [Электронный ресурс] : учебное пособие / Н. Вирт. — Электрон. дан. — Москва : ДМК Пресс, 2010. — 272 с. — Режим доступа: <a href="https://e.lanbook.com/book/1261">https://e.lanbook.com/book/1261</a> .	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
2	Павловская, Т. А. Программирование на языке С++ : учебное пособие / Т. А. Павловская. — 2-е изд. — Москва : ИНТУИТ, 2016. — 154 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/100409">https://e.lanbook.com/book/100409</a> (дата обращения: 08.09.2021). — Режим доступа: для авториз. пользователей.	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
3	Подбельский, В.В. Курс программирования на языке Си [Электронный ресурс] : учебник / В.В. Подбельский, С.С. Фомин. — Электрон. дан. — Москва : ДМК Пресс, 2012. — 384 с. — Режим доступа: <a href="https://e.lanbook.com/book/4148">https://e.lanbook.com/book/4148</a> .	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
<b>Дополнительная литература</b>		
1	Кауфман, В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс] / В.Ш. Кауфман. — Электрон. дан. — Москва : ДМК Пресс, 2010. — 464 с. — Режим доступа: <a href="https://e.lanbook.com/book/1270">https://e.lanbook.com/book/1270</a> .	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
2	Солдатенко, И.С. Практическое введение в язык программирования Си [Электронный ресурс] : учебное пособие / И.С. Солдатенко, И.В. Попов. — Электрон. дан. — Санкт-Петербург : Лань, 2018. — 132 с. — Режим доступа: <a href="https://e.lanbook.com/book/109619">https://e.lanbook.com/book/109619</a> . — Загл. с экрана.	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
3	Андрианова, А.А. Алгоритмизация и программирование. Практикум [Электронный ресурс] : учебное пособие / А.А. Андрианова, Л.Н. Исмагилов, Т.М. Мухтарова. — Электрон. дан. — Санкт-Петербург : Лань, 2019. — 240 с. — Режим доступа: <a href="https://e.lanbook.com/book/113933">https://e.lanbook.com/book/113933</a> — Загл. с экрана	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ

В.Бондарев

## Дополнительная литература

1. Вирт Н. Алгоритмы и структуры данных. — М.:Мир,1989. — 360с.
2. Керниган Б., Ритчи Д. Язык программирования СИ: Пер. с англ./Под ред. и с предисл. В.С. Штаркмана. — 2-е изд., перераб. и доп. — М. ; СПб. ; К. : Вильямс, 2006. — 272с.
3. Павловская Т. А. С/ С++. Программирование на языке высокого уровня : учеб. для студ. вузов, обуч. по напр. «Информатика и вычислительная техника» / Т. А. Павловская. — СПб.: Питер, 2009. — 461 с.
4. Павловская Т.А. С/С++. Структурное программирование: практикум / Т.А. Павловская, Ю.А. Щупак. — СПб.: Питер, 2004.—239 с.
5. Пильщиков В.Н. Сборник упражнений по языку Паскаль.-М.:Наука,1989. — 160с.

# Дополнительная литература

1. Абрамов В.Г. Введение в язык Паскаль : учеб. пособие для вузов по спец. "Прикладная математика" / В. Г. Абрамов, Н. П. Трифонов, Г. Н. Трифонова. - М. : Наука, 1988.
2. Немнюгин С.А. TURBO PASCAL: программирование на языке высокого уровня : учеб. для вузов / С. А. Немнюгин. - 2-е изд. - М. и др. : Питер, 2007. - 544 с.
3. Немнюгин С.А. Turbo Pascal: Практикум . - 2-е изд. – СПб.: Питер, 2005.— 268 с.



# Методические указания и пособия

- Методические указания к лабораторным работам по дисциплине «Алгоритмизация и программирование» для студентов дневной формы обучения направлений 09.03.02 — «Информационные системы и технологии» и 09.03.03 — «Прикладная информатика», часть 1 / Сост. В. Н. Бондарев, Т. И. Сметанина, А. К. Забаштанский. — Севастополь: Изд-во СевГУ, 2019. — 104 с.
- Методические указания к лабораторным работам по дисциплине «Алгоритмизация и программирование» для студентов дневной и заочной формы обучения направлений 09.03.02 — «Информационные системы и технологии» и 09.03.03 — «Прикладная информатика», часть 2 / Сост. В. Н. Бондарев, Т. И. Сметанина, А. К. Забаштанский. — Севастополь: Изд-во СевГУ, 2019. — 108 с.
- Методические указания к практическим занятиям по дисциплине «Алгоритмизация и программирование» для студентов направлений 09.03.02 — «Информационные системы и технологии» и 09.03.03 — «Прикладная информатика» / Сост. Т.И.Сметанина, В. Н. Бондарев. — Севастополь: Изд-во СевГУ, 2019. — 88с.
- Системы счисления и представление данных в памяти ЭВМ - Методические указания по выполнению расчетно-графической работы по дисциплине «Алгоритмизация и программирование» для студентов направлений 09.03.02 — «Информационные системы и технологии» и 09.03.03 — «Прикладная информатика» / Сост. Т.И.Сметанина, В. Н. Бондарев. — Севастополь: Изд-во СевГУ, 2019. — 32с.

**История развития ЭВМ.  
Понятие архитектуры компьютера.  
Принципы фон Неймана.**

# КРАТКАЯ ИСТОРИЯ РАЗВИТИЯ ВТ

## Ручной этап (период не установлен)

Приблизительно IV век до н.э. изобретение счётов (абака) — устройства, состоящего из набора костяшек, нанизанных на стержни. Абак позволял лишь запоминать результат, а все арифметические действия выполнял человек.



Рисунок 1. — Реконструкция римского абака

## Механический этап (с середины XVII века)

**1642 г. Блез Паскаль** создаёт первую механическую счетную машину, которая могла "запоминать" числа и выполнять элементарные арифметические операции (рис.2).



Рисунок 2. — Блез Паскаль (1623 - 1662гг.) и его суммирующая машина — «Паскалина»

**1670 г.** Создана арифметическая машина **Готфрида Лейбница** (1646 — 1716гг.). Первый в мире арифмометр, предназначенный для выполнения четырех действий арифметики.

## Механический этап (с середины XVII века)

**1823 г.** Английский ученый **Чарльз Бэббидж** (1791 —1871гг.) предложил проект **аналитической машины** (рис.3), в которую входили:

- «склад» для хранения чисел (память);
- «мельница» для операций над числами (процессор);
- устройство управления (процессор);
- устройства ввода/вывода.

Команды вводились с помощью перфокарт. Проект не был реализован.

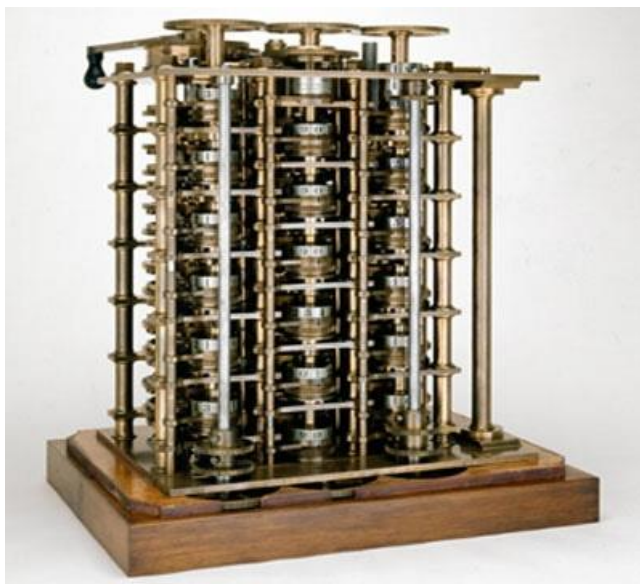


Рисунок 3. — Аналитическая машина Ч. Бэббиджа

## Механический этап (с середины XVII века)

**Ада (Августа) Лавлейс** (1815- 1852гг., дочь поэта Джорджа Байрона) — английский математик. Разработала основные принципы программирования **аналитической машины** Ч. Бэббиджа.



Составила первую в мире программу для этой машины (1843г.) Ввела в употребление понятия «**цикл**» и «**рабочая ячейка**», считается **первым программистом**.

Рисунок 4. — **Ада (Августа) Лавлейс**

## Электромеханический этап (с 90-х годов XIX века)

- **1888 г.** В США **Герман Холлерит** создаёт особое устройство – **табулятор**, в котором информация, нанесённая на перфокарты, расшифровывалась электрическими устройствами. Табулятор использовался для обработки результатов переписи населения в США.
- **1892 г.** Американский инженер **У. Барроуз** выпустил первый **коммерческий сумматор**.
- **1904—1906 гг.** Сконструированы электронные **диод и триод**.
- **1930 г.** Профессор Массачусетского технологического института (МТИ) **Ванневар Буш** построил **дифференциальный анализатор**, с появлением которого связывают начало современной компьютерной эры. Это была первая машина, способная решать сложные **дифференциальные уравнения**, которые позволяли предсказывать поведение движущихся объектов, например, самолета.



# Электронный этап (с 40-годов XX века)

## Довоенный период

**1936 г.** Английский математик **Алан Тьюринг** (1912 — 54гг) и независимо от него **Э. Пост** выдвинули и разработали концепцию **абстрактной вычислительной машины**. Они доказали принципиальную возможность решения автоматами любой проблемы при условии её алгоритмизации.



**Алан Тьюринг**

**1939 г.** Американец болгарского происхождения, профессор физики **Джон Атанасофф** создал **прототип вычислительной машины на базе двоичных элементов**.

**1941 г.** Немецкий инженер **Конрад Цузе** сконструировал **первый универсальный компьютер на электромеханических элементах**. Он работал с двоичными числами и использовал представление **чисел с плавающей запятой**.



## Послевоенный период

**1945 г.** Американский математик **Джон фон Нейман** (1903 – 57гг.) в отчёте "Предварительный доклад о машине Эдвак" сформулировал **основные принципы работы современных компьютеров.**



**Джон фон Нейман**

**1946 г.** Американцы **Преспер Эккерт** и **Джон Моучли** сконструировали **первый электронный цифровой компьютер ЭНИАК** (ENIAC – Electronic Numerical Integrator and Computer). Машина имела 20 тысяч электронных ламп и 1,5 тысячи реле, выполняя за одну секунду 300 умножений или 5000 сложений.

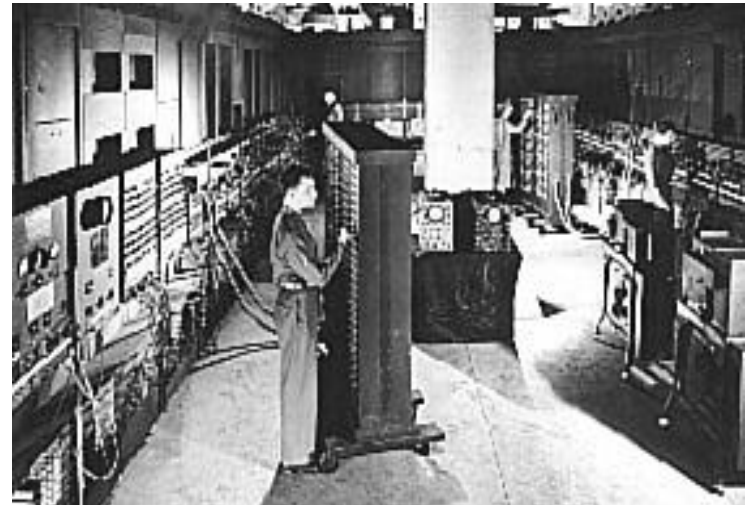


Рисунок 7. — П. Эккерт и Дж. Моучли, компьютер Эниак, 1946г.

## Послевоенный период

- **1948 г.** В американской фирме Bell Laboratories физики **Уильям Шокли, Уолтер Браттейн и Джон Бардин** создали **транзистор**. За это достижение им была присуждена Нобелевская премия.
- **1949 г.** Создан **первый полностью электронный компьютер ЭДВАК** (EDVAC – Electronic Discrete Variable Automatic Computer). В отличие от **ЭНИАК** (ENIAC), — это компьютер на двоичной, а не на десятичной основе.
- **1949 г.** В Англии под руководством **Мориса Уилкса** построен **первый в мире компьютер с хранимой в памяти программой EDSAC** (Electronic Delay Storage Automatic Computer).
- **1951 г.** В СССР , в Киеве в Институте электротехники под руководством академика **Сергея Алексеевича Лебедева (1902 – 1974гг.)** создан **первый в континентальной Европе электронный компьютер МЭСМ** (Малая Электронно-Счетная Машина).

## Разработка БЭСМ

- **1950 г.** С.А. Лебедев в Москве, в Институте точной механики и вычислительной техники (ИТМ и ВТ), начал разработку машины **БЭСМ** (**Б**ольшой **Э**лектронной **С**четной **М**ашины ).
- **1953 г.** БЭСМ принята Государственной комиссией в эксплуатацию. В 1956 г. доклад С.А. Лебедева о БЭСМ на международной конференции в Дармштадте произвел сенсацию — **БЭСМ была на уровне лучших американских машин и самой быстродействующей в Европе !**
- **1958 г.** С.А. Лебедев начал разработку **М-20** (цифра в названии указывала на ожидаемое быстродействие — 20 тыс. оп./с). Такой скорости вычислений тогда не имела ни одна машина в мире. **В 1958 г.** Государственная комиссия приняла М-20 и рекомендовала ее в серийное производство.

## Разработка БЭСМ

**1965 г.** ИТМ и ВТ начал проектирование полупроводниковой **БЭСМ-6**, которая обладала быстродействием 1 млн. оп./с. Главным конструктором БЭСМ-6 был С.А. Лебедев, заместителями — его ученики В.А. Мельников и Л.Н. Королев. В 1967 г. Государственная комиссия рекомендовала БЭСМ-6 к серийному производству. **По своим характеристикам БЭСМ-6 намного превосходила зарубежные разработки!**

БЭСМ-6 выпускалась Московским заводом Счетных аналитических машин (САМ) до 1985г. За разработку и внедрение БЭСМ-6 ее создатели были удостоены **Государственной премии.**



Академик Лебедев С.А.



БЭСМ-6

## Большие универсальные ЭВМ — мэйнфреймы

**1958 г.** Джек Килби из фирмы **Texas Instruments** создал первую интегральную схему (ИС). На базе интегральных схем началась разработка ЭВМ третьего поколения.

**1964 г.** Фирма **IBM** объявила о создании семейства ЭВМ **System 360** — первой **серии масштабируемых компьютеров**, впоследствии ставшей примером открытого стандарта, когда один производитель компьютерного оборудования мог произвести оборудование, совместимое с оборудованием другого производителя.

**1966 г.** В СССР была поставлена задача перехода к массовому производству **унифицированных ЭВМ — ЕС ЭВМ**, оснащённых большим количеством стандартизированного программного обеспечения и периферийного оборудования. За основу была принята архитектура системы **System 360**.

Первые **ЕС ЭВМ** появились в **1971 г.** Последние машины были выпущены в 1998г. (ЕС-1220). Всего было выпущено свыше 15 тыс. машин ЕС ЭВМ.

# Языки программирования

**1957 г.** Первое сообщение о языке программирования **Фортран** (Джон Бэкус).



**Джон Бэкус**

**1959 г.** Первое сообщение о языке **Алгол**, который надолго стал стандартом в области языков программирования.

**1965 г.** Дж. Кемени и Т. Курц в Дортмундском колледже (США) разработали язык программирования **Бейсик**.

**1970 г.** Швейцарец **Никлаус Вирт** разработал язык **Паскаль**.

**1972 г.** Деннис Ритчи из Bell Laboratories разработал язык **Си**.



**Никлаус Вирт**

**1983 г.** Фирмой **Borland** выпущен в продажу компилятор **Turbo Pascal**.

**1985 г.** Бьярн Страуструп из **Bell Laboratories** опубликовал описание созданного им объектно-ориентированного языка **С++**.



**Деннис Ритчи**

## Современный период (2010 - ...)

**2010г.** В национальном суперкомпьютерном центре в **Тяньзине (Китай)** создан суперкомпьютер **Tianhe-1A** производительностью **2.56 петафлопс ( $10^{15}$ )** на тесте Linpack (решение СЛАУ).

**2011г.** В **Японии** компанией **Fujitsu** построен самый быстродействующий в мире суперкомпьютер “**K computer**” с производительностью **10.51 петафлопс ( $10^{15}$ )**, содержащий **548352 ядра**.

**2012г. Sequoia** — проект суперкомпьютера, основанный на архитектуре Blue Gene. Разрабатывается компанией **IBM**, объявленная производительность **20 петафлопс**, содержащий **98304x16 ядер**.



## Современный период (2010- ...)

**2013-2015 гг. Tianhe-2 (Млечный путь) - 33 Петафлопс**, оперативная память - 1.4 Пб, внешнее ЗУ- 12.4 Пб, ОС - Kylin Linux. Китайский национальный университет оборонных технологий. Состоит из 16 тысяч узлов, каждый из которых включает 2 процессора Intel Xeon E5-2692 на архитектуре Ivy Bridge с 12 ядрами каждый (частота 2,2 ГГц) и 3 специализированных сопроцессора Intel Xeon Phi 31S1P (на архитектуре Intel MIC, по 57 ядер на ускоритель, частота 1,1 ГГц, пассивное охлаждение).

**2016г. Sunway TaihuLight - 93 Петафлопс** (теор. производительность 125,4 Петафлопс), Оперативная память - 1.3 Пб, ОС – Raise Linux . Национальный исследовательский центр параллельной вычислительной техники и технологий КНР. **В основе суперкомпьютера лежат новые китайские процессоры семейства ShenWei** — SW26010 с оригинальной 64-битной RISC-архитектурой, предположительно изготовленные по технологии 28 нм. Каждый процессор оснащен 260 ядрами, работает на частоте 1.45 ГГц и имеет производительность 3.06 терафлопс.



## Современный период (2010 - ...)



2019г. **Summit** — суперкомпьютер Ок-Риджской национальной лаборатории (США) вычислительной мощностью **122,3 ПФлопс**, самый мощный суперкомпьютер в мире по состоянию на 2019 год.

Комплекс занимает площадь около 520 м<sup>2</sup> и состоит из 4608 серверных узлов IBM Power Systems AC922, в общей сложности суперкомпьютер оснащён 9216 22-ядерных процессоров IBM POWER9 и 27 648 графических процессоров NVIDIA Tesla V100. Каждый узел содержит более 500 ГБ когерентной памяти (High Bandwidth Memory и DDR4 SDRAM), которая адресуется всеми CPU и GPU, плюс 800 ГБ энергонезависимой памяти, которая может использоваться как пакетный буфер или дополнительная память. В подсистеме охлаждения циркулирует 15 150 литров очищенной воды; потребляемая мощность системы в целом — 15 МВт (8100 среднестатистических жилых домов).

## Современный период (2010 - ...)



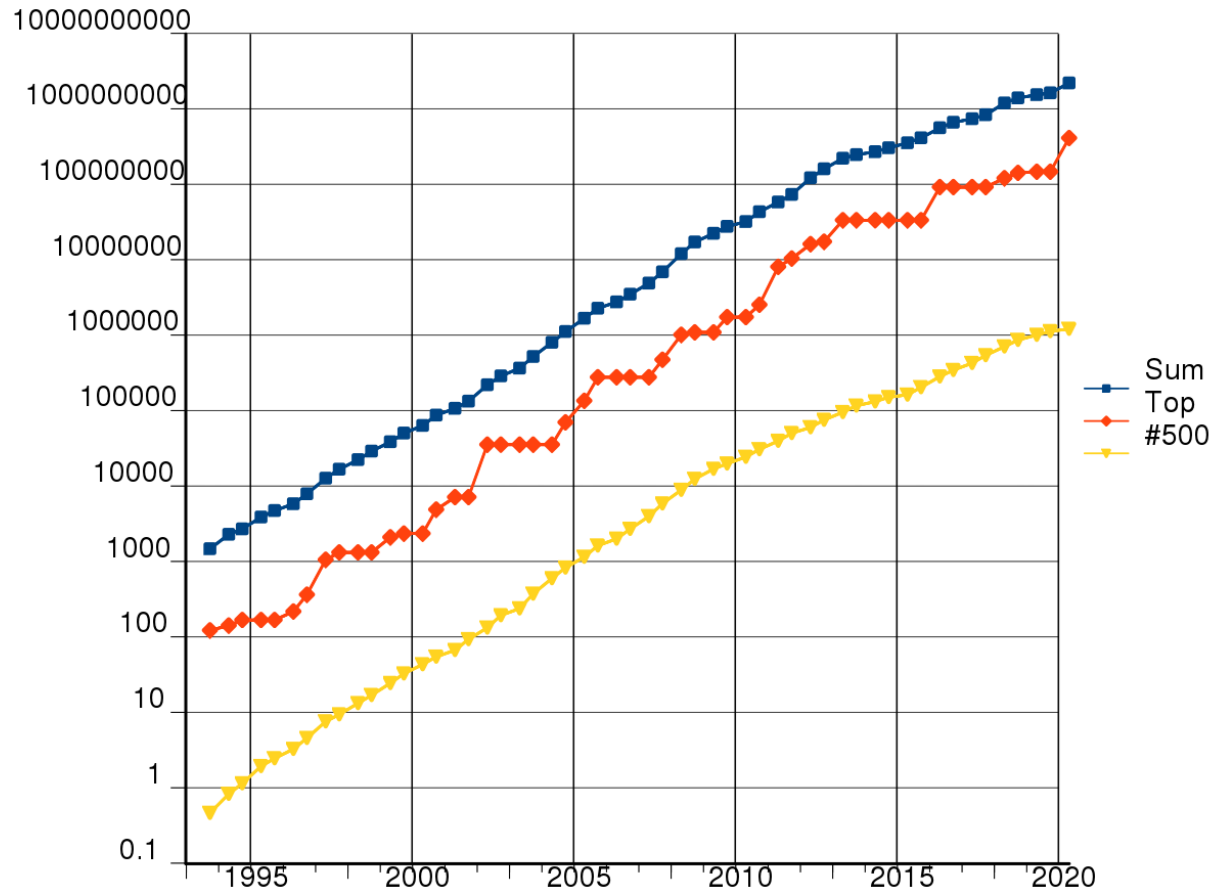
2020 г. Суперкомпьютер Фугаку (Fugaku, Институт физико-химических исследований, Япония ) показал в тесте High Performance Linpack (HPL) результат **415,5 петафлопса**, в 2,8 раза опередив занимающую второе место американскую систему IBM Summit.

Fugaku используются 48-ядерные однокристальные системы Fujitsu A64FX SoC на архитектуре Arm. В операциях с одинарной или меньшей точностью, которые часто используются в задачах ИИ, включая машинное обучение, пиковая производительность Fugaku превышает 1 эксафлопс. Операционная система – Linux (RHEL), потребляемая мощность 28,3 МВт.

Эксперты назвали этот суперкомпьютер Fugaku лучшим еще в трех категориях: при использовании промышленности; при использовании в сфере искусственного интеллекта; при анализе больших цифр. Примечательно, что до этого ни один компьютер в мире не занимал первое место сразу в четырех номинациях из шести.

В.Бондарев

# Рост производительности суперкомпьютеров, Гигафлопс ( $10^9$ )



2020г. - 1000 Петафлопс =  $10^{18}$  флопс = 1 Экса(за) флопс !

# Рост производительности суперкомпьютеров

Распределение суперкомпьютеров в списке Топ500 по странам (июнь 2018 года)

страна	количество
Китай	206
США	124
Япония	36
Великобритания	22
ФРГ	21
Франция	18
Голландия	9
Южная Корея	7
Ирландия	7
Канада	6
Австралия	5
Индия	5
Италия	5
Польша	4
Россия	4
Саудовская Аравия	4
Швейцария	3
Швеция	3
Сингапур	2
Испания	2
ЮАР	1
Тайвань	1
Норвегия	1
Бразилия	1
Новая Зеландия	1

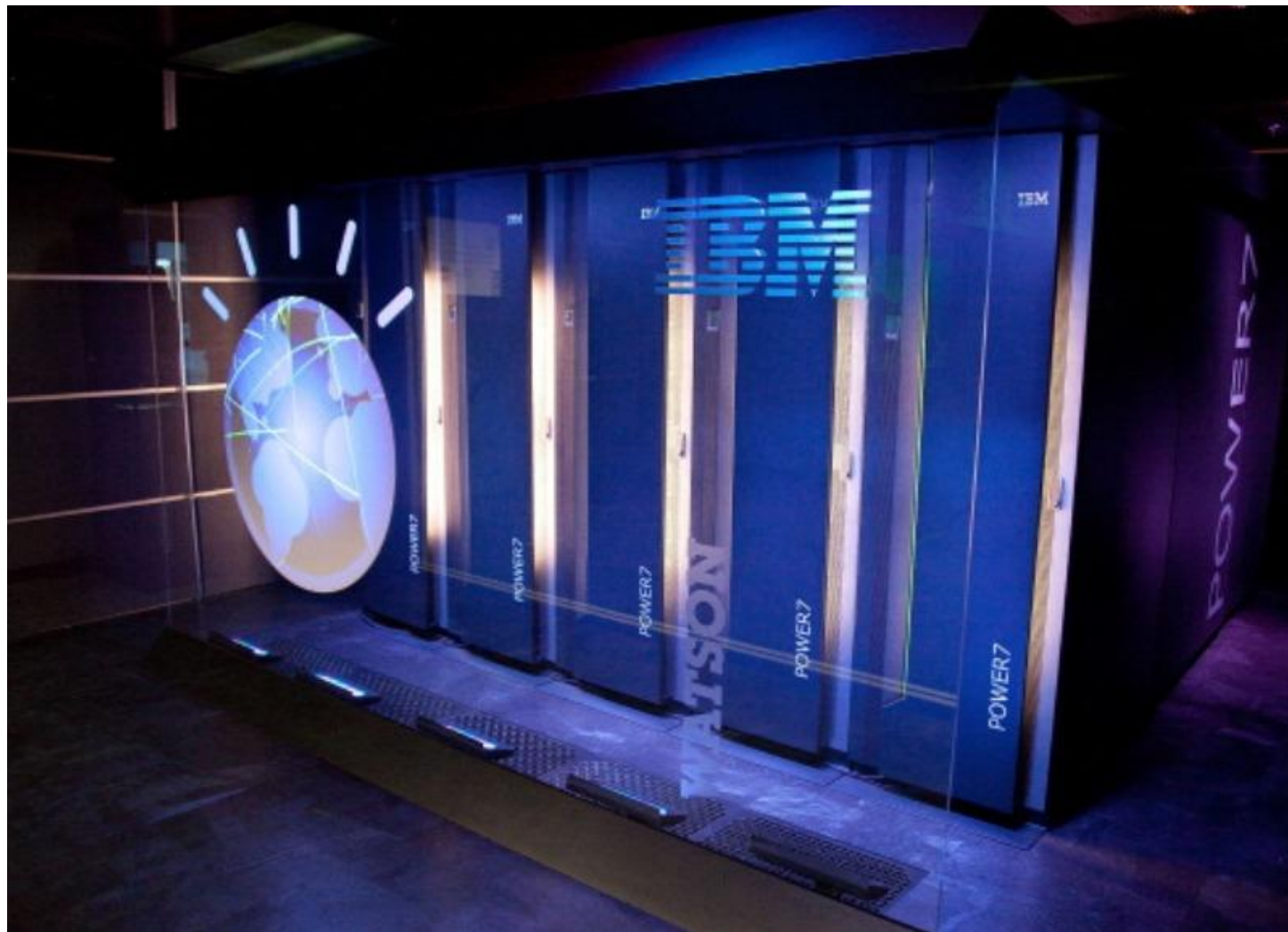
# Поколения компьютеров

**Поколение ЭВМ** – период развития ВТ, отмеченный относительной стабильностью архитектуры и технических решений. **Смена поколений** связана с переходом на новую элементную базу.

Поколение. Годы.	Элементная база	Быстродействие (оп/сек.)	Программное обеспечение (ПО).	Модели ЭВМ, ПО.
1. 1946-55е	Электронные лампы	$(10-20) 10^3$	Машинные коды	БЭСМ-2, Стрела
2. 1955-65	Транзисторы	$(0.2 - 1) 10^5$	+ Алгоритмические языки	БЭСМ – 6, М220
3. 1965-75	Интегральные схемы (ИС)	$10^7$	+ Операционные системы (ОС), Мультипрограммирование	ЕС 1010 -1060, МИР-2, СМ ЭВМ
4. 1975-90е	Большие интегральные схемы (БИС)	$10^9$	+ Сетевое ПО, Параллельное программирование	ЕС1015-1065, СМ-1420, Эльбрус 2, ПЭВМ
5. 1990- ?	Сверхбольшие интегральные схемы (СБИС)	$1000 \times 10^{12}$ ( 1 петафлопс)	+ Интеллектуальное ПО с дружественным интерфейсом,  ПО для объединения ресурсов компьютеров	IBM Watson  Суперкомпьютеры



# IBM суперкомпьютер Watson с СИИ



# IBM суперкомпьютер Watson с СИИ

- **Основная задача** — понимать вопросы, сформулированные на ЕЯ и находить на них ответы в базе данных. Назван в честь основателя IBM Томаса Уотсона
- **Участие в «Jeopardy !»**
- В феврале 2011 года для проверки возможностей Уотсона он принял участие в телеигре **Jeopardy**. Уотсон одержал победу, получив 1 миллион долларов.
- **Платформа** Уотсон состоит из 90 серверов Power7 750, каждый из которых содержит по 4 восьмиядерных процессора POWER7. Суммарная оперативная память Уотсона более 15 терабайт.
- **Система** имела доступ к 200 миллионам страниц структурированной и неструктурированной информации объемом в 4 терабайта, включая полный текст Википедии. Во время игры Уотсон не имел доступа к интернету.

# Проекты Blue Brain и ASIMO

Проект **Blue Brain** предполагает построение **искусственной модели мозга** путем реинжиниринга мозга млекопитающих до молекулярного уровня. Моделирование выполняется на суперкомпьютере ( **Blue Gene**) с производительностью 0.5 Пфлопс с помощью программного пакета **NEURON** (Michael Hines). Основная задача заключается в построении симулятора **биологической нейронной сети млекопитающих**.

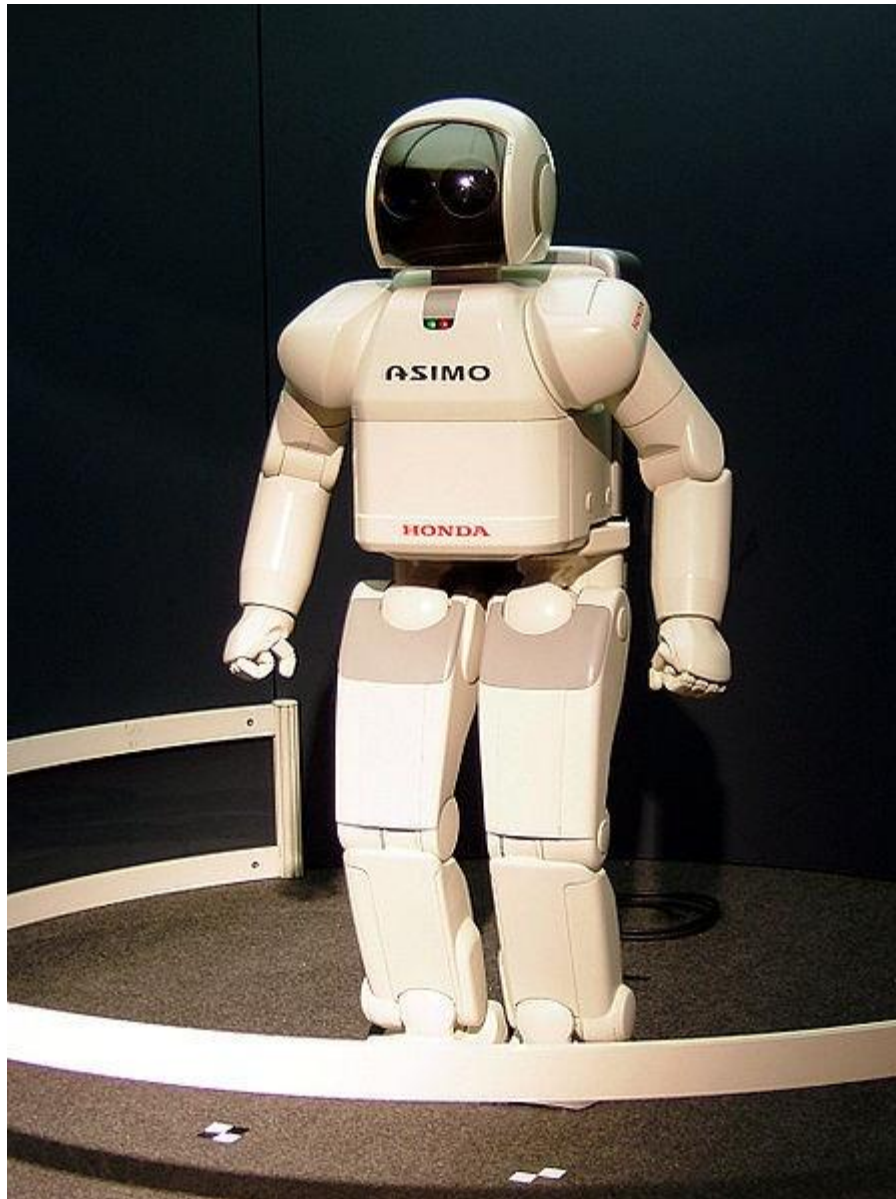
**ASIMO (アシモ *ashimo*)** – это **человекоподобный робот**, созданный компанией **Honda**, который может перемещаться на **двух ногах** со скоростью 6 км/час. Имеет **подсистему ИИ**, которая обладает следующими характеристиками:

- обнаруживает движущиеся объекты с помощью камеры;
- интерпретирует перемещение рук собеседника (распознавание жестов);
- оценивает своё положение для безопасного перемещения;
- идентифицирует источники звуков, реагирует на голосовые вопросы движением головы или словесными ответами;
- распознает лица (даже, если робот или человек перемещаются).

ASIMO имеет **сетевые подключения**, которые обеспечивают его применение в качестве гида для работы с посетителями офисов.



# ASIMO - человекоподобный робот с ИИ



# Понятие архитектуры компьютера

Различают понятия **архитектуры** компьютера и **организации** компьютера.

**Архитектура компьютера** – это логическая организация, структура и ресурсы компьютера с точки зрения программиста.

**Архитектура** определяет принципы действия, информационные связи и взаимное соединение основных **логических устройств** компьютера (процессора, ЗУ, периферийных устройств). Она предполагает описание: системы команд компьютера, методов адресации, форм представления данных в памяти, пользовательских возможностей программирования. При этом физическая организация устройств компьютера не учитывается

**Организация компьютера** или **структурная организация** – это описание конкретной реализации архитектуры, т.е. воплощение её в виде конкретных функциональных элементов и их взаимосвязей.

*Прикладному программисту довольно часто требуется знание архитектуры компьютера, реже его организации и весьма редко – схемы компьютера.*

# Принципы архитектуры Джона фон Неймана

В настоящее время применяются две основные архитектуры ЭВМ: **принстонская** (Дж. фон Нейман) и **гарвардская** (Говард Эйкен).

**Основные принципы архитектуры фон-Неймана:**

**1. Принцип двоичного кодирования информации в компьютере.**

Для представления данных и команд используется двоичная система счисления;

**2. Принцип программного управления компьютером.**

Компьютер управляется программой, состоящей из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности;

**3. Принцип хранения программы в памяти компьютера и однородности памяти.**

Как программы (команды), так и данные хранятся в одной и той же памяти. Над командами также можно выполнять действия, как и над данными;

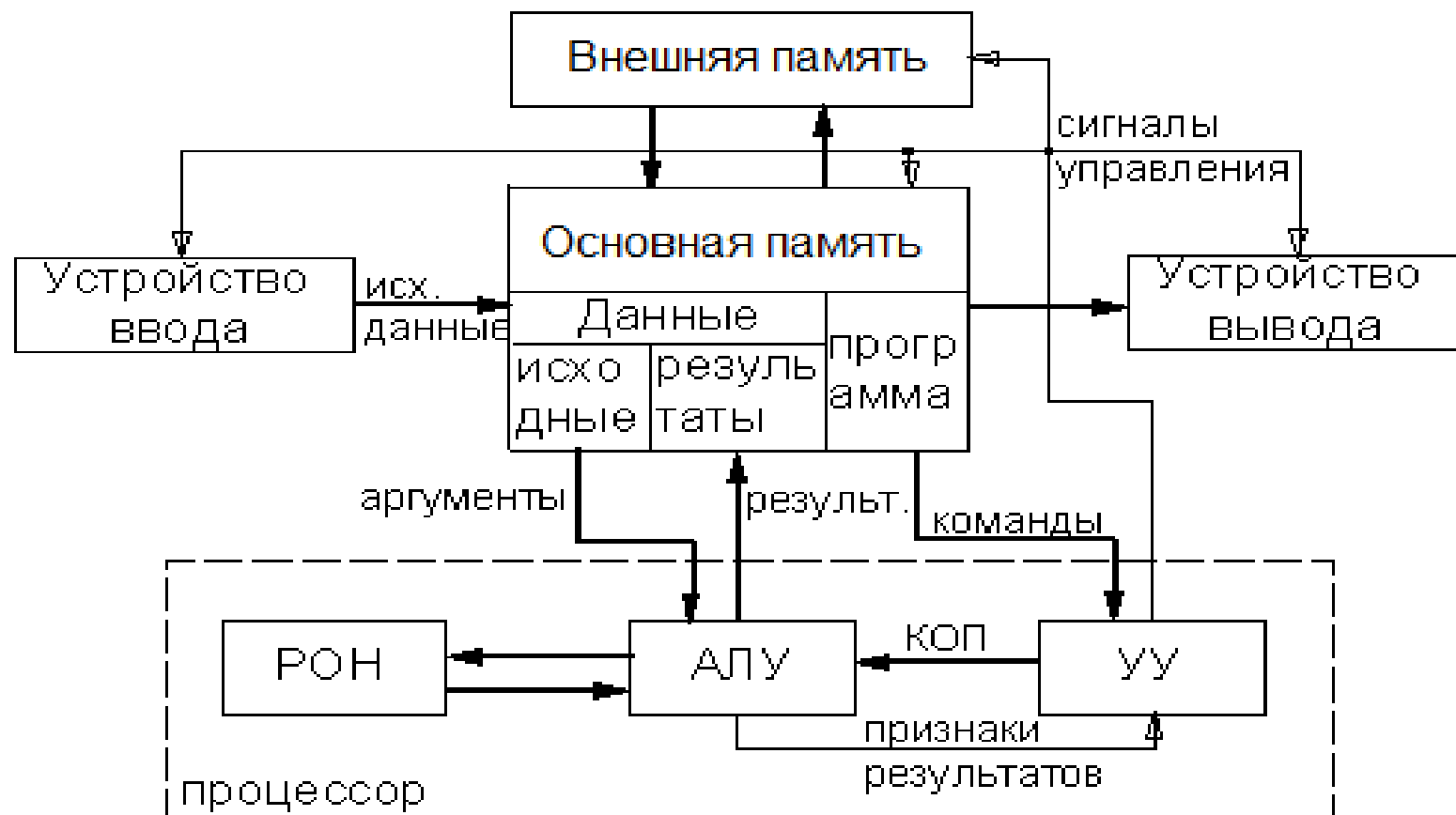
**4. Принцип адресности.**

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

**Гарвардская архитектура** – раздельное хранение программ и данных.

# Архитектура компьютеров фон Неймана

Типовая фон-Неймановская машина содержит: арифметико-логическое устройство (АЛУ), устройство управления (УУ), память, устройства ввода-вывода.



**РОН** – регистры общего назначения

# УВВ и Память

**Устройства ввода-вывода (УВВ)** предназначены для ввода-вывода исходных данных и программ в память и из памяти компьютера.

**Память** компьютера предназначена для хранения программ, исходных и промежуточных данных, результатов. Она подразделяется на **внутреннюю** и **внешнюю**. Внутренняя память состоит из: *основной памяти, регистров и кеш-памяти*.

**Основная память (ОП)** компьютера подразделяется на **постоянную (ПЗУ, ROM)** и **оперативную (ОЗУ, RAM)**. **ПЗУ** – это **энерго-независимая** память, а **ОЗУ** – **энерго-зависимая**, т.е. при отключении питания информация, хранящаяся в **ОЗУ**, будет утеряна.

**Внешняя память** предназначена для длительного хранения больших объёмов данных и программ.

Память компьютера разбита на **байты**. Номер байта называется **адресом**.

Максимальный номер байта называется **емкостью памяти**.

Если **внутренняя память** характеризуется **высоким быстродействием** и относительно **небольшой емкостью**, то **внешняя**, наоборот, характеризуется **более низким быстродействием** и значительно **большей емкостью**.

# Процессор

**Процессор** состоит из **РОН**, **АЛУ**, **УУ** и предназначен для выполнения арифметических и логических операций над данными и управления всеми устройствами машины.

**РОН** – **быстродействующая память** небольшой емкости. Позволяет повысить скорость выполнения операций в процессоре за счет уменьшения неэффективных пересылок данных из процессора в **ОП** и обратно.

**УУ** обеспечивает синхронную работу всех узлов машины и выполняет следующие функции:

- 1) извлекает из **ОП** очередную команду программы и дешифрирует её;
- 2) передает **код операции (КОП)** дешифрованной команды в **АЛУ**;
- 3) формирует адреса для извлечения необходимых данных из памяти;
- 4) отдает приказание **АЛУ** исполнить требуемую операцию.

**АЛУ** предназначено для выполнения арифметических и логических операций. По результатам выполнения операций **АЛУ** формирует специальные **признаки**, которые анализируются в **УУ**. Используя эти признаки **УУ**, может выбрать не следующую по порядку команду, а **передать управление команде**, которая обеспечит выполнение альтернативной ветви алгоритма в соответствии с возникшей ситуацией.

# Архитектура системы команд

Элементарная операция, выполняемая за один шаг, называется **командой**.  
Полный перечень команд, который может выполнять компьютер, называется **системой команд**.

Типовая команда задаёт:

- ✓ операцию, которую необходимо выполнить;
- ✓ адреса исходных данных (операндов);
- ✓ адрес памяти, где необходимо сохранить результат выполнения операции.

Команда состоит из 2-х частей: **операционной** и **адресной**.

В зависимости от количества адресов, указываемых в адресной части, различают **1-но адресные, 2-х адресные и 3-х адресные команды**.



**КОП** – код операции; **А1, А2, А3** – адреса ячеек оперативной памяти, где размещаются исходные данные и результат

## Выполнение команд

Команды программы выполняются **последовательно**, начиная с первой. Адрес очередной команды хранится в **счетчике команд (СК)**, входящем в **УУ**.

**Процесс выполнения команды** состоит из следующих шагов:

- 1) из ячейки памяти, на которую указывает **СК**, выбирается очередная команда; содержимое **СК** увеличивается на длину команды;
- 2) команда передается в **регистр команд УУ**, где дешифрируются адреса;
- 3) по сигналам **УУ** операнды из памяти передаются в **АЛУ** ;
- 4) **УУ** дешифрирует **КОП** и передает его в **АЛУ** для выполнения операции;
- 5) результат операции записывается в память по соответствующему адресу (АЗ) или остается в процессоре;
- 6) повторение шагов 1) – 5) , пока не будет считана команда “стоп”.

В некоторых случаях выполняется **не следующая по порядку команда**, а в зависимости от ***признаков результатов*** операции происходит **передача управления** другой команде, т. е. нарушается **линейный принцип** выполнения программ, и появляется возможность **разветвлений**.



## Пример программы на машинном коде

Рассмотрим гипотетическую ЭВМ. Предположим, что наша ЭВМ должна уметь выполнять следующие операции:

Операция	КОП
+ (сложение)	01
- (вычитание)	02
* (умножение)	03
/ (деление)	04

Пусть требуется написать программу для вычисления выражения:

$$z = \frac{x \cdot y}{x + y - A}$$

Вычислять выражение будем по шагам:

$$r1 = x \cdot y$$

$$r2 = x + y$$

$$r2 = r2 - A$$

$$z = r1 / r2$$

Здесь  $r1$ ,  $r2$  – переменные, используемые для временного хранения промежуточных результатов;  $A$  – константа; знак “=” – оператор присваивания.

## Пример программы на машинном коде

Чтобы связать имена переменных с адресами ячеек памяти, где будут храниться значения переменных, создается **таблица адресов**. Процесс её создания называется **распределением памяти**.

Имя (идентификатор) переменной	Адрес ячейки памяти
x	100
y	101
r1	102
r2	103
A	104
z	105

Пусть начальный адрес размещения в памяти самой программы – 200.

Запишем программу в **машинных кодах** для 3-х адресной машины :

Адрес команды	КОП	A1	A2	A3
200	03	100	101	102
201	01	100	101	103
202	02	103	104	103
203	04	102	103	105