

# Детализация ответов на вопросы к зачету по курсу История развития вычислительной техники и языков программирования

## 1. Принципы фон Неймана в организации работы ЭВМ.

- **Принцип двоичного кодирования:** вся информация кодируется в двоичном виде.
- **Принцип программного управления:** программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.
- **Принцип однородности памяти:** программы и данные хранятся в одной и той же памяти.
- **Принцип адресности:** память состоит из пронумерованных ячеек; процессору в любой момент времени доступна любая ячейка.

## 2. Поколения ЭВМ: элементная база, основные характеристики (память, тактовая частота, программные решения)

- I поколение (1945 - 1955)  
электронно-вакуумные лампы
- II поколение (1955 - 1965)  
транзисторы
- III поколение (1965 - 1980)  
интегральные микросхемы
- IV поколение (1980 – по настоящее время...)  
большие и сверхбольшие интегральные схемы (БИС и СБИС)

Параметры сравнения	Поколения ЭВМ			
	первое	второе	третье	четвертое
Период времени	1946 - 1959	1960 - 1969	1970 - 1979	с 1980 г.
Элементная база (для УУ, АЛУ)	Электронные (или электрические) лампы	Полупроводники (транзисторы)	Интегральные схемы	Большие интегральные схемы (БИС)
Основной тип ЭВМ	Большие		Малые (мини)	Микро
Быстродействие (операций в сек)	$10^4$	$10^5$	$10^6$	$10^7 - 10^9$
Основные устройства ввода	Пульт, перфокарточный, перфоленточный ввод	Добавился алфавитно-цифровой дисплей, клавиатура	Алфавитно-цифровой дисплей, клавиатура	Цветной графический дисплей, сканер, клавиатура

Основные устройства вывода	Алфавитно-цифровое печатающее устройство (АЦПУ), перфоленточный вывод		Графопостроитель, принтер	
Внешняя память	Магнитные ленты, барабаны, перфоленты, перфокарты	Добавился магнитный диск	Перфоленты, магнитный диск	Магнитные и оптические диски, CD и DVD
Ключевые решения в ПО	Автокоды, Ассемблеры	Универсальные языки программирования, трансляторы Пакетные операционные системы, оптимизирующие трансляторы	Интерактивные операционные системы, структурированные языки программирования	Дружественность ПО, сетевые операционные системы
Режим работы ЭВМ	Однопрограммный	Пакетный	Разделения времени	Персональная работа и сетевая обработка данных
Цель использования ЭВМ	Научно-технические расчеты	Технические и экономические расчеты	Управление и экономические расчеты	Телекоммуникации, информационное обслуживание
Названия ЭВМ	ENIAC, Урал-1; МЭСМ; ЭБЭСМ-2	IBM; БЭСМ-6	IBM-360; ЕС-ЭВМ	IBM PC/XT/AT Pentium; Athlon etc

### 3. Что ожидалось от машин V поколения.

- Цель – создание суперкомпьютера с функциями искусственного интеллекта
- Основные черты проекта:
  - обработка *знаний* с помощью логических средств (язык Пролог)
  - сверхбольшие базы данных
  - использование параллельных вычислений
  - распределенные вычисления
  - голосовое общение с компьютером
  - постепенная замена программных средств на аппаратные
- Проблемы:
  - идея саморазвития системы провалилась
  - неверная оценка баланса программных и аппаратных средств
  - традиционные компьютеры достигли большего
  - ненадежность технологий
  - развитие Интернета – новая распределенная модель хранения данных
  - израсходовано 50 млрд. йен
  - технические средства приближаются к пределу быстродействию
  - сложность программного обеспечения приводит к снижению надежности
- Перспективы:
  - квантовые компьютеры
  - эффекты квантовой механики
  - параллельность вычислений
  - 2006 – компьютер из 7 кубит

- оптические компьютеры («замороженный свет»)
- биокомпьютеры на основе ДНК
- химическая реакция с участием ферментов
- 330 трлн. операций в секунду

#### 4. Стандартизация вычислительной техники. System/360: семь основных принципов.

##### *New Product Line*

- центральный процессор должен с равным успехом использоваться для научных и деловых вычислений;
- все члены будущего семейства должны быть способны работать с *одним и тем же набором периферийных устройств*;
- для центрального процессора была выбрана гибридная технология, допускавшая использование микросхем и навесной монтаж дискретных элементов (к тому времени уже существовали микросхемы, но они еще не обладали достаточной надежностью);
- в научных и бизнес-вычислениях должен использоваться *один и тот же язык программирования высокого уровня*;
- между всеми членами семейства должна сохраняться *программная совместимость*;
- адресация в семействе должна обеспечивать доступ к 16 млн. символов, а в перспективе – к 2 млрд.;
- минимальной единицей представления данных будет **8-битовый байт**.

#### 5. Достоинства выбранных форматов стандарта System/360

- 8-битное поле обеспечивало *эффективное кодирование десятичных данных*, т.к. в деловых отчетах они встречаются вдвое чаще, чем алфавитно-цифровые.
- Короткие форматы обеспечивают *скорость и эффективность кодирования*, длинные – *точность*. 48-бит был самым распространенным форматом, а формат 32/64 давал возможность выбора между точностью и скоростью.
- В выбранных 32- и 64-битных форматах *на знак отводился 1 бит*, на *порядок – 7*, а оставшиеся разряды отводились под мантиссу.

#### 6. История развития ЭВМ в СССР. БЭСМ-6.

Научные школы развития средств вычислительной техники

- *Сергея Лебедева*, ставшего идеологом ЭВМ с высоким быстродействием;
- *Иссака Брука*, занимавшегося разработкой малых и управляющих ЭВМ;

- *Бориса Рамеева*, который до конца 60-х годов возглавлял направление, связанное с разработкой универсальной ЭВМ.

Типичные представители советских ЭВМ I-III поколений:

К *первому* поколению советских компьютеров относятся ламповые ЭВМ - "БЭСМ-2", "Стрела", "М-2", "М-3", "Минск", "Урал-1", "Урал-2", "М-20".

Ко *второму* поколению советских компьютеров относятся полупроводниковые малые ЭВМ "Наири" и "Мир", средние ЭВМ для научных расчетов и обработки информации со скоростью 5-30 тысяч операций в секунду "Минск-2", "Минск-22", "Минск-32", "Урал-14", "Раздан-2", "Раздан-3", "БЭСМ-4", "М-220" и управляющие ЭВМ "Днепр", "ВНИИЭМ-3", а также сверхбыстродействующая БЭСМ-6 с производительностью *1 млн операций в секунду*.

Компьютеры *третьего* поколения на интегральных микросхемах появились в СССР во второй половине 1960-х годов. Были разработаны Единая Система ЭВМ (ЕС ЭВМ) и Система Малых ЭВМ (СМ ЭВМ) и организовано их серийное производство. Как уже указывалось выше, эта система представляла собой клон американской системы IBM/360.

*Роль ЕС ЭВМ в развитии отечественных компьютеров неоднозначна.*

На начальном этапе появление ЕС ЭВМ *привело к унификации компьютерных систем, позволило установить начальные стандарты программирования и организовывать широкомасштабные проекты, связанные с внедрением программ.*

Ценой этого было *повсеместное свёртывание собственных оригинальных разработок и попадание в полную зависимость от идей и концепций фирмы IBM, далеко не самых лучших по тому времени.*

## **7. Основные архитектурные решения, применяемые в микропроцессорах**

- Принстонская/гарвардская архитектуры
- Конвейерная архитектура
- Суперскалярная архитектура
- полный и урезанный наборы команд (CISC/RISC-процессоры)
- Многоядерность
- Кэширование
- Векторность

Первым суперскалярным компьютером считается CDC 6600, разработанный С. Креем в 1964 г., а первыми суперскалярными массовыми микропроцессорами – SuperSPARC (1992) и Pentium (1993).

Практически все современные процессоры являются такими, они оснащены несколькими блоками выборки, декодирования, арифметических операций и пр.

## 8. CISC- и RISC-процессоры.

**CISC (Complex Instruction Set Computing)** – концепция проектирования процессоров, характеризующаяся полным набором команд. Объединение нескольких машинных операций в одной команде позволило непосредственно поддерживать языки высокого уровня, увеличить плотность кода и уменьшить обращения к памяти. В начале 1960-х это позволило добиться большой экономии на памяти и большей производительности при программировании на ассемблере, т.к. языки высокого уровня, такие как Алгол или Fortran, были не всегда доступны.

**RISC (Reduced Instruction Set Computing)** – вычисления с сокращённым набором команд. Концепция разработана независимо в IBM (1975) и университете Беркли (1980) с целью преодоления недостатков микропроцессоров CISC. Архитектура основана на статистическом анализе используемых команд и операндов, развитой регистровой архитектуре, *отказе от микрокода и отказе от использования сложных команд в пользу нескольких простых.*

Архитектура RISC позволила существенно упростить структуру процессора и получить большую производительность, чем CISC, поэтому с 1990-х гг. почти все процессоры являются RISC или RISC-подобными (Power), либо это CISC-процессоры с RISC-ядром (x86).

## 9. Эволюция операционных систем (ОС).

Периоды в развитии операционных систем

- 1 период. Ламповые машины. **Операционных систем нет.**
- 2 период. Компьютеры на основе транзисторов. **Пакетные операционные системы.**
- 3 период. Компьютеры на основе ИС. **Первые многозадачные ОС.**
- 4 период. Персональные компьютеры. **Классические, сетевые и распределенные системы.**
- 5.период. ОС для мобильных устройств.

## 10. Пакетные ОС

- С середины 50-х годов начался следующий период в эволюции вычислительной техники, связанный с переходом на новую элементную базу – полупроводниковые элементы. **Применение транзисторов вместо часто перегоравших электронных ламп повысило надежность компьютеров и существенно увеличило время безотказной работы.**
- Теперь на них можно было возложить выполнение практически важных задач. Снизилось потребление электроэнергии, уменьшились размеры, снизилась также стоимость эксплуатации обслуживания вычислительной техники. Началось использование компьютеров коммерческими фирмами. Одновременно наблюдается бурное развитие алгоритмических языков (Lisp, COBOL, ALGOL-60, PL-1 и т.д.).

- Происходит **разделение персонала**, ранее ответственного за весь цикл разработки и использования компьютеров, **на программистов и операторов**, специалистов по эксплуатации и разработчиков вычислительных машин.
- Изменяется **сам процесс прогона программ**. Теперь пользователь приносит программу с входными данными в виде колоды перфокарт и указывает необходимые ресурсы. Такая колода получает название **задания**.
- Оператор загружает задание в память машины и запускает его на исполнение. Полученные выходные данные печатаются на принтере, и пользователь получает их обратно через некоторое время. **Смена запрошенных ресурсов вызывает приостановку выполнения программ, в результате процессор часто простаивает.**
- Для повышения эффективности использования компьютера задания с похожими ресурсами **начинают собирать вместе**, создавая **пакет заданий**. Так появляются первые **системы пакетной обработки**, которые просто автоматизируют запуск одной программы из пакета за другой и тем самым увеличивают коэффициент загрузки процессора.
- При реализации систем пакетной обработки был разработан **формализованный язык управления заданиями**, с помощью которого программист сообщал системе и оператору, какую работу он хочет выполнить на вычислительной машине.
- **Системы пакетной обработки стали прообразом современных операционных систем**, они были первыми системными программами, предназначенными для управления вычислительным процессом.

## 11. Мультипрограммные ОС: режим разделения памяти и времени

Дальнейшее повышение эффективности использования процессора было достигнуто с помощью **мультипрограммирования** – одновременного выполнения нескольких программ, относящихся к разным задачам.

1. При возникновении задержки при выполнении одной из программ, например из-за поиска на магнитной ленте участка, где хранятся исходные данные, **её выполнение прерывается** и осуществляется **переход к диспетчер-программе**, которая передаёт управление следующей программе.
2. Когда операция поиска (или ввода- вывода) заканчивается, процессор возвращается к выполнению первой программы. При этом каждая программа загружается в свой **участок оперативной памяти, называемый разделом**, и не влияет на выполнение другой программы.

Такая мультипрограммность называется **РЕЖИМ РАЗДЕЛЕНИЯ ПАМЯТИ**

- **Логическим расширением систем мультипрограммирования стали системы разделения времени.**
  1. В них процессор переключается между задачами не только на время операций ввода-вывода, но и просто по прошествии определенного времени.
  2. Эти переключения происходят так часто, что пользователи могут взаимодействовать со своими программами во время их выполнения, то есть интерактивно.

3. В результате появляется возможность одновременной работы нескольких пользователей на одной компьютерной системе.

- Внедрение **механизма виртуальной памяти**, при которой в оперативную память загружался только фрагмент программы, который необходимо в данный момент выполнять, а остальная часть программы остается на диске, позволило создавать иллюзию неограниченности оперативной памяти.

- Это ослабило ограничения на количество одновременно работающих пользователей. В системах разделения времени, применяя электронно-лучевой монитор и клавиатуру, **стало возможным производить отладку программы в интерактивном режиме.**

## 12. История развития и возможности ОС UNIX

ОС UNIX (1969): Ситуация в сфере технологии разработок ОС.

- Эклектичный характер разработки (смешанный из разных стилей и идей) систем
- Сложность и громоздкость
- Полная несовместимы между собой.
- Разработка и отладка ОС затягивались. Документирование часто не велось.

**Создать новую систему часто было проще, чем пытаться отладить старую.**

Идея: разработать собственную ОС, которая отличалась бы максимальной простотой и минимальными размерами

UNIX обладала **файловой иерархической** системой, предоставляла возможность использования **сетевыми возможностями**, поставлялась с **компилятором C**, обладала **интерпретатором команд**, поддерживала написание скриптов и пр.

С 1974 г. система стала **бесплатно распространяться** среди университетов и академических учреждений, что способствовало **популяризации Unix и языка C**, а также переносу на разные аппаратные платформы. В результате ОС Unix стала синонимом «Открытой системы».

## 13. История развития и возможности ОС LINUX

Linux (1990) В 1990 г. студент университета Хель-синки **Линус Торвальдс**, изучавший Unix – купил книгу Танненбаума вместе с Minix, скачал и установил соответствующие дополнения.

В августе 1991 г. он объявил о создании новой операционной системы – Linux. Система была выложена **в свободный доступ** и стала распространяться по лицензии GNU GPL (Универсальная общедоступная лицензия)

Особенности Linux

- *Linux не является клоном Unix.*
- *Linux можно считать первой настоящей POSIX-системой.*
- Linux создавался на машине с процессором i386 для архитектуры Intel и, первоначально, – только для нее.

- Поэтому соответствие стандартам требовалось не для переносимости Linux самого по себе, а для обеспечения **возможности компиляции в этой ОС всего ранее созданного программного ассортимента** для Unix и POSIX-совместимых систем вообще.
- С другой стороны, следование стандарту сделало возможным перенос Linux-софта на другие Unix-подобные платформы.
- Линус создал метод разработки масштабных проектов **Open Sources**.

## 14. Тенденции в развитии современных ОС: развитие пользовательского интерфейса, сетевые возможности, мультипрограммность, дружелюбность и т.д.

1. Современные операционные системы развиваются в направлении:
  - поддержки новых **сетевых** технологий,
  - **многopotочности** и **многоядерности**.
2. Развиваются **проекты с открытым исходным кодом**.
3. Развивается **виртуализация**, обеспечивающая возможность выполнения приложения в среде другой ОС.
4. В конце 2000-х гг. появляется поддержка
  - **облачных вычислений**,
  - большого успеха добиваются и операционные **системы для мобильных платформ**.

## 15. Технология облачных вычислений

**Облачные вычисления** – технология, которая позволяет использовать дополнительные ресурсы, находящиеся на других серверах. Пользователь системы загружает все необходимые программы и данные на серверы компании, или использует **ПО** предлагаемое компанией-поставщиком услуги. Сервисы облачных вычислений предполагает управление программным обеспечением **Cloud Computing** через обычные и привычные любому пользователю веб-браузеры. Облачные вычисления - динамично развивающаяся технология использования информационной инфраструктуры.

***Человек Подключенный станет Человеком Арендующим.***

Достоинства технологии «облако»

- Распределение нагрузки в системе облачных вычислений происходит автоматически.
- Благодаря этой особенности системы облачных вычислений осуществляется более быстрое вхождение в программное обеспечение.
  - Кроме того, значительно снижаются затраты на электроэнергию и обслуживание,
  - увеличивается скорость обработки информации,
  - экономится пространство на жестком диске,
  - нет необходимости наращивать мощность компьютера.



## 16. Понятие транслятор: компилятор и интерпретатор.

Транслятор – программа, переводящая текст программы с языка высокого уровня в язык машинных кодов.

- ✗ ассемблеры переводят программу, записанную на языке ассемблера, в программу на машинном языке;
- ✗ интерпретаторы - это транслятор, который производит *пооператорную* обработку и выполнение исходного кода программы;
- ✗ компиляторы – транслятор, который преобразует *всю программу* в модуль на машинном языке, после чего программа записывается в память компьютера и лишь потом исполняется)

## 17. Языки программирования низкого уровня

*Машинные коды.* Для упрощения записи двоичная система записывалась в виде восьмеричной или шестнадцатеричной, но вводить коды нужно было все равно в системе исчисления с машинным основанием. Контроль ввода и конечного результата, а также поиск ошибок, осуществлялся по контрольным лампочкам на панели управления, либо по распечаткам.

*Ассемблер – это мнемоническая запись кодов центрального процессора.* Программу, написанную на ассемблере, переводит специальная программа-транслятор, заменяя каждую ассемблерную команду на соответствующий двоичный код. Набор полученных кодов полностью соответствует исходному набору ассемблерных команд, он *может быть непосредственно загружен в память машины и выполнен процессором.* Однако, поскольку каждый процессор использует только свой набор кодов, *такая программа не может быть непосредственно выполнена на машине с другим типом процессора.*

## 18. Языки программирования высокого уровня.

- Fortran (1956), научные и инженерные расчеты.
- COBOL (1959), для управления и бизнеса.
- ALGOL (1958/1960), для записи алгоритмов.
- PL/1 (1962), разработан для System/360.
- BASIC (1964), для записи простых программ в учебных целях.

## 19. Универсальные языки программирования.

- Pascal (1970), улучшение Алгола и задачи обучения.
- C (1973), для системного программирования.
- Ada (1980), для встроенных систем реального времени.
- Lisp (1958), Prolog (1972), обработка символьной информации и систем искусственного интеллекта.