ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Лекция № 01 — Вводная

Преподаватель: Поденок Леонид Петрович, 505а-5

+375 17 293 8039 (505а-5)
+375 17 320 7402 (ОИПИ НАНБ)
prep@lsi.bas-net.by

ftp://student:2ok*uK2@Rwox@lsi.bas-net.by

Кафедра ЭВМ, 2023

2023.02.14

Оглавление

Определения....3
Иерархическое представление вычислительной системы....5
  Обобщенная структура вычислительной системы (взгляд IT-специалиста)....6
  Абстрактная архитектура вычислительной системы....6
  Многоуровневая схема вычислительной системы (теория)....7
Понятие ресурса в ОС....8
  Концепция виртуализации ресурса....9
  Виртуальная память....11
  Виртуализация оборудования....11
  Виртуальные инструкции ввода/вывода....12
  Виртуальные инструкции параллельной обработки....14
Аппаратура. Концептуальный состав ЭВМ....15
  Архитектура фон-Неймана. Принципы....17
  Гарвардская архитектура....19
  Как это все работает (Цикл выполнения команды)....20
  Память....21
  Периферийные устройства....21
Операционная система (ОС)....23
Система управления файлами (СУФ)....24
Интерфейсные оболочки пользователя (Shell)....25
  Переносимость....26
Система программирования....27
  Жизненный путь программы....29
  Связь между файлами программы и утилитами....31
  Поддержка периферийных устройств и доступ к аппаратуре....33
Нестрогая классификация ОС....34
  Мэйнфреймы и малые ЭВМ....34
  Настольные системы....36
  Мультипроцессорные системы....36
  Распределенные системы....37
    Системы Клиент-сервер....38
    Системы Pear-to-Pear....39
Кластерные системы....39
ОС реального времени....39

Определения

ОС — довольно расплывчатое понятие, опирающееся на определенные традиции, параметры оборудования, размеры программ, реализующих определенные функции, распределение ресурсов:

- загрузка пользовательских программ в оперативную память и их исполнение;
- работа с устройствами долговременной памяти, такими как магнитные диски, ленты, оптические диски и т.д. Как правило, ОС управляет свободным пространством на этих носителях и структурирует пользовательские данные;
- предоставляет более или менее стандартный доступ к различным устройствам ввода/вывода, таким как терминалы, модемы, печатающие устройства;
- предоставляет пользовательский интерфейс. Часть систем ограничивается командной строкой, в то время как другие поддерживают графический пользовательский интерфейс;

Существуют ОС, функции которых этим и исчерпываются.

Одна из систем такого типа — DOS (DR DOS, IBM PC-DOS, MS DOS).

Более развитые ОС предоставляют также следующие возможности:

- параллельное исполнение нескольких задач;
- распределение ресурсов компьютера между задачами;
- организация взаимодействия задач друг с другом;
- взаимодействие пользовательских программ с нестандартными внешн. устройствами;
- организация межмашинного взаимодействия и разделения ресурсов;
- защита системных ресурсов, данных и программ пользователя, исполняющихся процессов и самой себя от ошибочных и зловредных действий пользователей и их программ.

Иерархическое представление вычислительной системы

Предложено Дейкстрой

Вычислительная система рассматривается как многослойная структура из нескольких функциональных уровней.

Каждый уровень определяется данными, выполняемыми функциями и результатами.

Самый нижний — аппаратура (микроархитектура). Каждый более вышестоящий для выполнения своих функций может запрашивать сервис более низкого уровня.

Результат выполнения функций нижнего уровня передается на более высокий уровень.

При этом имеет место изоляция несмежных уровней друг от друга.

Обобщенная структура вычислительной системы (взгляд IT-специалиста)

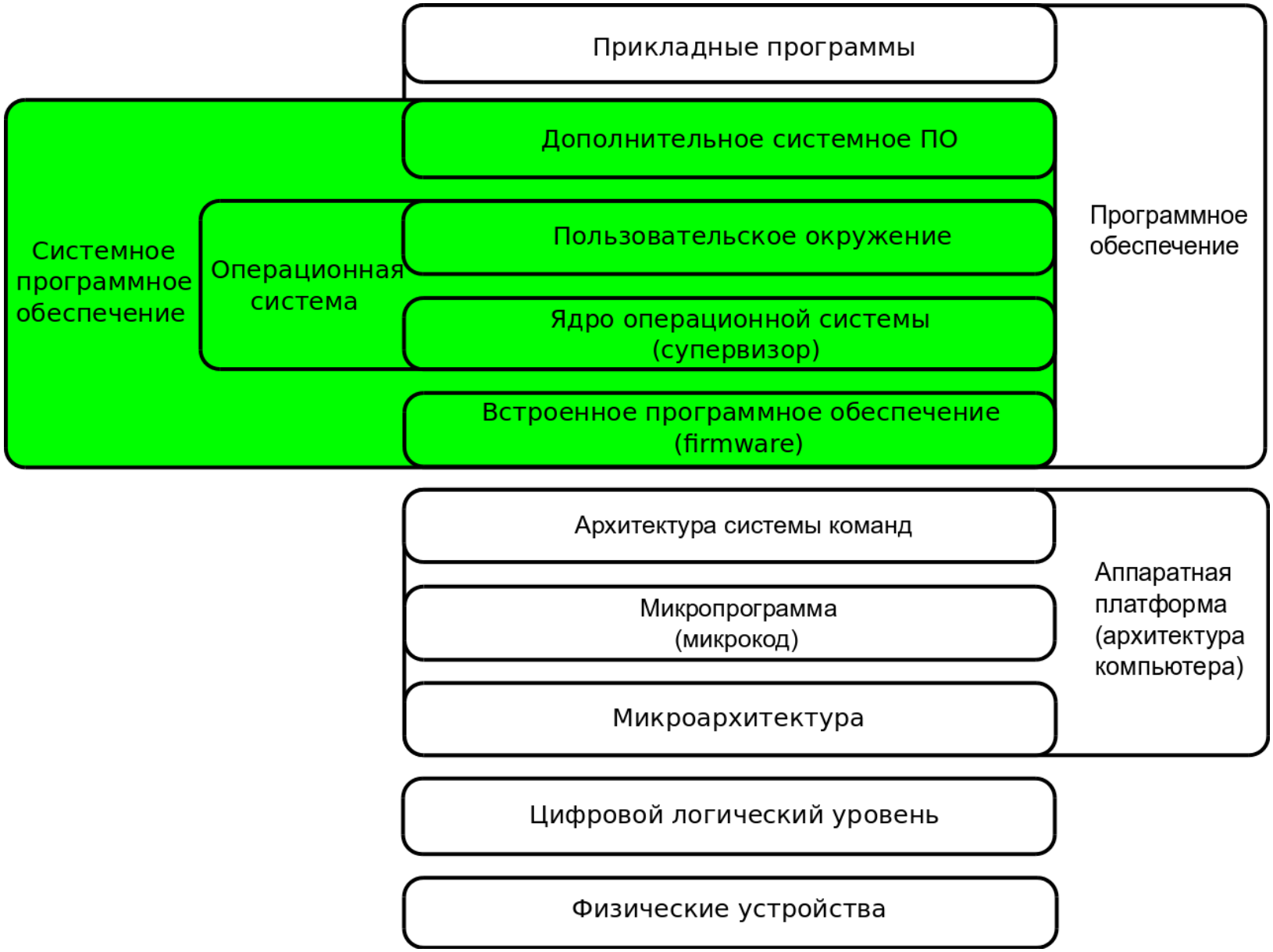
Наш курс, в основном, имеет отношение к слоям, которые на схеме обозначены как «**СИСТЕМНЫЙ ИНТЕРФЕЙС**» и «**Операционная система**»

Абстрактная архитектура вычислительной системы

Уровень 3	**OSM** — Уровень машины операционной системы	Операционная система
Уровень 2	**ISA** — Уровень архитектуры набора инструкций	Машинный код
Уровень 1	**MA** — Уровень микроархитектуры	Микропрограммы или оборудование (процессор)

Каждый верхний уровень наследует некоторое подмножество команд нижнего и добавляет новые. На уровне MA и ISA команды различных систем существенно различаются.

Многоуровневая схема вычислительной системы (теория)



Понятие ресурса в ОС

Общая цель всех ОС — обеспечение эффективного и бесконфликтного распределения ресурсов ЭВМ между пользователями.

Ресурс — всякий потребляемый объект, независимо от формы его существования, обладающий некоторой практической ценностью для потребителя.

Вычислительный ресурс — возможности, обеспечиваемые компонентами вычислительной системы, расходуемые или занимаемые в процессе её работы.

К числу основных ресурсов вычислительных систем могут быть отнесены такие ресурсы, как процессоры, основная память, таймеры, наборы данных, устройства хранения данных, коммуникационные устройства.

Ресурсы различаются по запасу выделяемых единиц ресурса и бывают в этом смысле **исчерпаемые** и **неисчерпаемые**.

Исчерпаемость ресурса, как правило, приводит к конфликтам в среде потребителей. Для урегулирования конфликтов ресурсы должны распределяться между потребителями по правилам, в наибольшей степени их удовлетворяющим.

Вычислительную систему можно представить как ограниченное множество функциональных элементов, обладающих потенциальными возможностями выполнения с их помощью или над ними действий, связанных с обработкой, хранением или передачей данных. Ресурсы распределяются между процессами.

Концепция виртуализации ресурса

Одним из важнейших свойств ресурса является «реальность существования».

В этом смысле ресурсы разделяют на физические и виртуальные (мнимые).

Под физическим понимают ресурс, который реально существует и при распределении его между пользователями обладает всеми присущими ему физическими характеристиками (физическая память, доступ к физическому каналу).

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию вычислительных процессов, выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска нескольких операционных систем на одном компьютере.

Каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения).

Хостовая операционная система или гипервизор управляет предоставлением таких логических ресурсов из общего пула, доступного на уровне оборудования.

Также могут быть подвергнуты виртуализации сети передачи данных (сокеты), сети хранения данных, платформенное и прикладное программное обеспечение.

Виртуальный ресурс схож многими своими характеристиками с некоторым физическим, но по многим свойствам и отличен — это некоторая модель физического ресурса.

Виртуальный ресурс не существует реально в том виде, в котором он проявляет себя пользователю.

Свойство виртуализации ресурсов — одной из важнейших при построении систем управления ресурсами. По значимости — это одна из важнейших концепций при построении современных ОС.

Операционная система — с точки зрения программиста это программа, которая добавляет множество новых инструкций и особенностей помимо тех, которые обеспечиваются уровнем архитектуры набора инструкций. Это:

- виртуальная память;
- виртуализация оборудования;
- виртуализация инструкций ввода/вывода;
- виртуализация инструкций параллельной обработки.

Виртуальная память

1) Оверлеи и ручное распределение памяти. Мастер-оверлей, распределение и загрузка ==> Тупик. ==> Автоматическая реализация подгрузки кода.

2) Сегментация — множество независимых логических адресных пространств.

3) Страничная организация — код и данные хранятся на диске в виде страниц фиксированного размера. Часть страниц находится в ОП. По мере того, как требуются отсутствующие страницы, они подгружаются в ОП.

4) Кэширование. Програма в ОП разбивается на блоки фиксированного размера , часть которых находится в кэш-памяти. Отличается от (4) организацией.

Виртуализация оборудования

Одновременно на конкретной аппаратной архитектуре может выполняться несколько операционных систем (интернет-хостинг, облачные вычисления, ...).

Поддержка виртуализации на аппаратном уровне.

<table>
<tr><td>Прилож. A</td><td>Прилож. B</td><td>Прилож. B</td><td>Прилож. C</td><td>Прилож. D</td><td>Прилож. E</td><td>Прилож. E</td><td>Прилож. F</td></tr>
<tr><td colspan="2">ОС_1</td><td colspan="2">ОС_1</td><td colspan="2">ОС_1</td><td colspan="2">ОС_1</td></tr>
<tr><td colspan="2">Аппаратная поддержка виртуальной машины</td><td colspan="2">Аппаратная поддержка виртуальной машины</td><td colspan="2">Аппаратная поддержка виртуальной машины</td><td colspan="2">Аппаратная поддержка виртуальной машины</td></tr>
<tr><td colspan="8">Гипервизор (программный компонент)</td></tr>
<tr><td colspan="8">Аппаратная архитектура и периферийные устройства физического компьютера</td></tr>
</table>

Виртуальные инструкции ввода/вывода

Набор команд уровня ОС содержит большую часть команд уровня ISA и добавляет несколько очень важных новых. Работа с вводом выводом на уровне ISA представляет потенциальную угрозу и слишком трудоемка.

Например, чтобы считать или записать информацию на дискету, необходимо

- запустить двигатель вращения дискеты;
- управлять шаговым двигателем перемещения головки;
- следить за индикатором присутствия дискеты;
- выбрать номер блока на диске;
- выбрать дорожку;
- выбрать номер сектора на дорожке;

...

Все эти функции берет на себя операционная система.

Файлы — очень важная абстракция виртуального ввода-вывода.

Последовательность байт (октетов — ASN.1), записанных на УВВ.

Если УВВ — устройство хранения (диск), файл можно прочитать обратно, в противном случае невозможно (принтер). На диске может храниться много файлов различной длины, содержащих те или иные данные.

Файловая абстракция позволяет организовать виртуальный ввод-вывод.

На уровне ОС файл является просто последовательностью байтов. Вся остальная структуризация выполняется на уровне прикладных программ.

- открытие;
- запись/чтение;
- закрытие.

Реализация виртуализации выполняется в виде файловой системы (от перфокарт до каталогов). Команды управления каталогами:

- создание файла и включение в каталог;
- удаление файла из каталога;
- переименование;
- изменение статуса защиты.

Виртуальные инструкции параллельной обработки

Путь повышения быстродействия — параллельные вычисления.

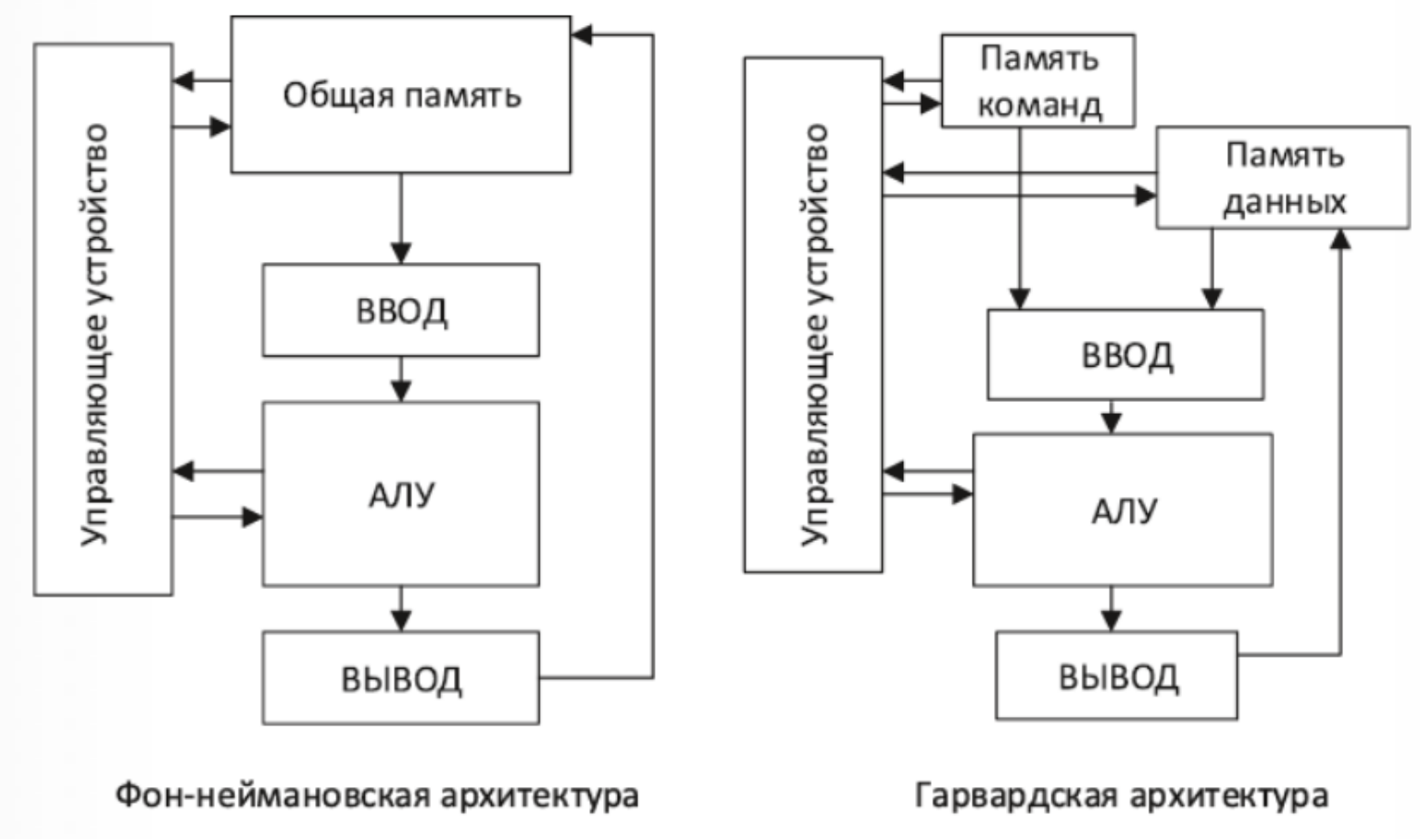
ВС может иметь несколько процессоров, процессоры несколько ядер, но на всех не хватает — совместное использование процессора несколькими процессами.

Новые инструкции — системные вызовы:

- формирование процесса (клонирование текущего или исходное состояние);
- прерывание процесса;
- остановка/продолжение;
- синхронизация.

Аппаратура. Концептуальный состав ЭВМ

Любой IBM PC-совместимый компьютер представляет собой реализацию суперпозиции фон-Неймановской и гарвардской архитектур.



- блок управления;
- арифметико-логическое устройство (АЛУ);
- память;
- устройства ввода/вывода.

Данная архитектура реализует концепцию хранимой в памяти программы — прорам-ма и данные хранятся в одной и той же памяти.

Блок управления и АЛУ определяют действия, выполняемые ЭВМ, и составляют цен-тральный процессор (ЦП, CPU).

ЦП последовательно выбирает из памяти команды и исполняет их. Этот тип архитек-туры называется архитектура, управляемая потоком команд.

Адрес очередной команды в памяти задается счетчиком адреса в блоке управления.

Архитектура фон-Неймана. Принципы

Принцип однородности памяти

Команды и данные хранятся в одной и той же памяти и внешне в памяти неразличимы. Распознать их можно только по способу использования.

Это значит, что одно и то же значение в ячейке памяти может использоваться и как данные, и как команда, и как адрес в зависимости от способа обращения к нему.

Это позволяет производить над командами те же операции, что и над числами, и, соответственно, открывает ряд возможностей, например модификацию команд.

Принцип адресности

Структурно основная память состоит из пронумерованных ячеек, причём процессору в произвольный момент доступна любая ячейка. Двоичные коды команд и данных разделяются на единицы информации, называемые словами, и хранятся в ячейках памяти, а для доступа к ним используются номера соответствующих ячеек — адреса.

Принцип программного управления

Все вычисления, предусмотренные алгоритмом решения задачи, представлены в виде программы, состоящей из последовательности управляющих слов — команд.

Каждая команда предписывает выполнение некоторой операции из реализуемого набора операций.

Команды программы хранятся в последовательных ячейках памяти вычислительной машины и выполняются в естественной последовательности, то есть в порядке их положения в программе.

При необходимости, с помощью специальных команд, эта последовательность может быть изменена. Решение об изменении порядка выполнения команд программы принимается либо на основании анализа результатов предшествующих вычислений, либо безусловно.

Принцип двоичного кодирования

Согласно этому принципу, вся информация, как данные, так и команды, кодируются двоичными цифрами 0 и 1.

Каждый тип информации представляется двоичной последовательностью и имеет свой формат (int, float …).

В формате команды в простейшем случае можно выделить два поля:

- поле кода операции;
- поле адресов/операндов.

КОП	Операнды

Гарвардская архитектура

Гарвардская архитектура — архитектура ЭВМ, отличительными признаками которой являются:

- хранилище инструкций и хранилище данных представляют собой разные физические устройства;
- канал инструкций и канал данных так же физически разделены.

В гарвардской архитектуре характеристики устройств памяти для инструкций и памяти для данных не обязательно должны быть одинаковыми.

Ширина слова, протокол доступа, технология реализации и структура адресов памяти могут различаться.

В некоторых системах инструкции могут храниться в памяти только для чтения (ПЗУ, ROM), в то время как для сохранения данных обычно требуется память с возможностью чтения и записи (ОЗУ, RAM).

В некоторых системах требуется значительно больше памяти для инструкций, чем для данных, что обуславливает различную ширину шин адреса памяти инструкций и данных.

Как это все работает (Цикл выполнения команды)

При выполнении команды, связанной с обращением к памяти, процессор должен выполнить как минимум пять операций, перечисленных ниже:

1. **Выборка команды**. Блок управления извлекает команду из памяти, копирует ее во внутреннюю память микропроцессора и увеличивает значение счетчика команд на длину этой команды.
2. **Декодирование команды**. Блок управления определяет тип выполняемой команды, пересылает указанные в ней операнды в АЛУ и генерирует электрические сигналы управления АЛУ, соответствующие типу выполняемой операции.
3. **Выборка операндов**. Если в команде используется операнд, расположенный в памяти, блок управления инициирует операцию по его выборке из памяти.
4. **Выполнение команды**. АЛУ выполняет указанную в команде операцию, сохраняет полученный результат в заданном месте и обновляет состояние флагов, по значению которых программа может судить о результате выполнения команды.
5. **Запись результата в память**. Если результат выполнения команды должен быть сохранен в памяти, блок управления инициирует операцию сохранения данных в памяти.

Память

Оперативная память — массив ячеек со смежными адресами. Реализуется на микросхемах динамической памяти.

Для повышения производительности между памятью и процессором располагается сверхоперативная память небольшого объема (до трех уровней кеш-памяти)

Энергонезависимая память ПЗУ (ROM BIOS).

Процессор, память и необходимые элементы взаимодействия их между собой и с другими устройствами называют центральной частью или ядром ЭВМ.

Периферийные устройства

Все остальные компоненты, не вошедшие в ядро.

Их иногда называют устройствами ввода-вывода (УВВ), что не совсем точно.

- устройства хранения данных;
- коммуникационные устройства;
- устройства ввода-вывода.

Процессор, память и периферийные устройства взаимодействуют между собой с помощью шин и интерфейсов, аппаратных и программных.

Стандартизация интерфейсов делает архитектуру компьютеров открытой.

Некоторые термины

Биты — квант информации (1 или 0)
Байты — квант адресуемой памяти
Слова — группа байт (наследие 8086/88)
BigEndian — адресуется старший байт (коммуникации, не x86)
LittleEndian — адресуется младший байт (x86)

Не путать с MSB/LSB — Most/Lest Significant Bit

Ячейки памяти — служат для хранения, поддерживают операции записи и чтения.
Порты — преобразование двоичной информации в электрические сигналы.
Регистры **— широкое многоуровневое понятие.**

Операционная система (ОС)

Операционная система — комплекс управляющих и обрабатывающих программ, который

- выступает как интерфейс между аппаратурой компьютера и пользователем;
- предназначен для эффективного и безопасного использования ресурсов ВС.

Любой из компонентов прикладного программного обеспечения обязательно работает под управлением ОС.

Ни один из компонентов программного обеспечения, за исключением самой ОС, не имеет непосредственного доступа к аппаратуре компьютера.

Пользователи взаимодействуют со своими программами через интерфейс ОС.

Любые их команды, прежде чем попасть в прикладную программу, сначала проходят через ОС.

Система управления файлами (СУФ)

Назначение — организация доступа к данным, организованным в виде файлов. Как правило, все современные ОС имеют соответствующие системы управления файлами.

Однако выделение СУФ в отдельную категорию представляется нецелесообразным, поскольку ряд ОС позволяет работать с несколькими файловыми системами (либо с одной из нескольких, либо сразу с несколькими одновременно).

В этом случае говорят о монтируемых файловых системах, и в этом смысле они самостоятельны.

Система управления файлами (СУФ) не существует сама по себе – обычно она разработана для работы в конкретной ОС и с конкретной файловой системой.

Для работы с файлами, организованными в соответствии с некоторой файловой системой, для каждой ОС должна быть разработана соответствующая система управления файлами.

Интерфейсные оболочки пользователя (Shell)

Классический интерфейс для всех ОС — CLI.

Наиболее мощный и эффективный.

Для удобства взаимодействия с ОС могут использоваться дополнительные интерфейсные оболочки. Их основное назначение – либо расширить возможности по управлению ОС, либо изменить встроенные в систему возможности.

Графические интерфейсные оболочки:

- X Window в системах семейства UNIX (опция);
- Presentation Manager (PM — оконный интерфейс IBM OS/2) (опция);
- Work Place Shell (объектный рабочий стол IBM OS/2) (опция);
- Windows (Microsoft);
- MAC OS (Apple);
- Foton (QNX 4.x) (опция).

Переносимость

Ряд операционных систем могут организовывать выполнение программ, созданных для других операционных систем.

OS/2 — MS-DOS, Windows 3.x (нативное выполнение).

Linux — Windows 32/64

Free BSD — Linux

Windows NT — MS-DOS, OS/2 1.x, Windows 3.x.

Система программирования

К этой категории относятся системные программы, предназначенные для разработки программного обеспечения

ассемблеры — компьютерные программы, осуществляющие преобразование программы в форме исходного текста на языке ассемблера в машинные команды в виде объектного кода;

трансляторы — программы или технические средства, выполняющее трансляцию программы;

компиляторы — программы, переводящие текст программы на языке высокого уровня, в эквивалентную программу на машинном языке.

интерпретаторы — программы (иногда аппаратные средства), анализирующие команды или операторы программы и тут же выполняющие их;

компоновщики (редакторы связей) — программы, которые производят компоновку — принимают на вход один или несколько объектных модулей и собирают по ним исполнимый модуль;

препроцессоры исходных текстов — это компьютерные программы, принимающие данные на входе, и выдающие данные, предназначенные для входа другой программы, например, такой, как компилятор;

отладчики (англ. debugger) — модули среды разработки или отдельные программы, предназначенные для поиска ошибок в программах;

текстовые редакторы — компьютерные программы, предназначенные для создания и изменения текстовых файлов, а также их просмотра на экране, вывода на печать, поиска фрагментов текста и т. п.;

специализированные редакторы исходных текстов — текстовые редакторы для создания и редактирования исходного кода программ. Специализированный редактор исходных текстов может быть отдельным приложением, или быть встроен в интегрированную среду разработки;

библиотеки подпрограмм — сборники подпрограмм или объектов, используемых для разработки программного обеспечения;

редакторы графического интерфейса.

генераторы лексических анализаторов — программы, позволяющие по формальным описаниям лексем, генерировать программы лексического анализа текстов.

генераторы компиляторов — программы, позволяющие по формальным описаниям грамматик, генерировать трансляторы.

системы документирования программ.

системы управления версиями.

Жизненный путь программы

Обычно программа проходит нескольких шагов:

- текст на алгоритмическом языке;
- объектный модуль;
- загрузочный модуль;
- бинарный образ в памяти.

Трансляция программы — преобразование программы, представленной на одном из языков программирования, в программу на другом языке. Транслятор обычно выполняет также диагностику ошибок, формирует словари идентификаторов, выдаёт для печати текст программы и т. д.

Виды трансляции:

- компиляция;
- интерпретация;
- динамическая компиляция.

Компилятор (compiler) — транслятор, преобразующий исходный код с какого-либо языка программирования на машинный язык.

Процесс компиляции, как правило, состоит из нескольких этапов:

- лексический анализ;
- синтаксический анализ;
- семантический анализ;
- создание на основе результатов анализов промежуточного кода;
- оптимизация промежуточного кода;
- создание объектного кода, в данном случае машинного.

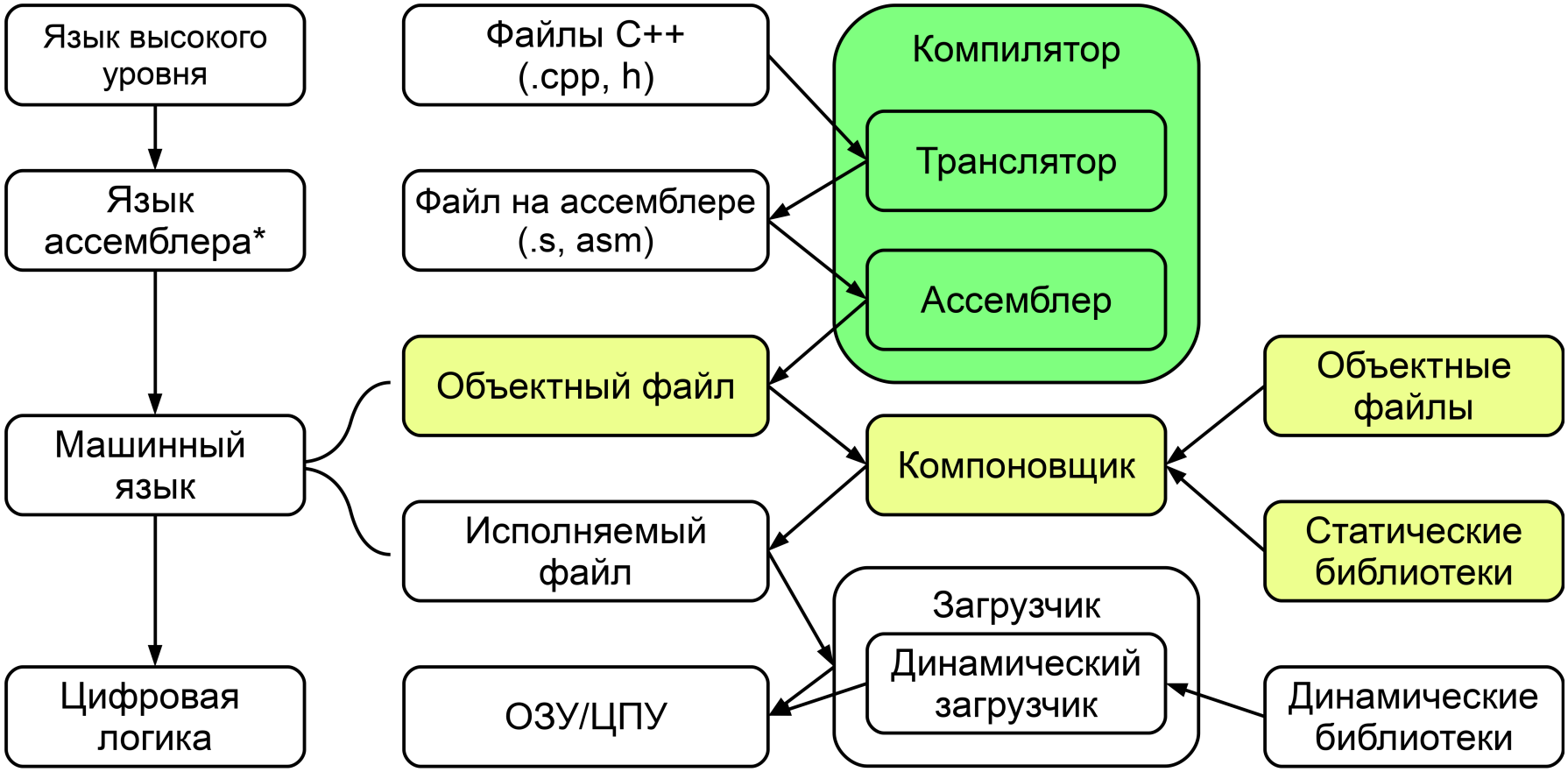
Используемые программой адреса в каждом конкретном случае могут быть представлены различными способами. Например, адреса в исходных текстах обычно символические.

Компилятор связывает эти символические адреса с перемещаемыми адресами (такими как N байт от начала модуля).

Загрузчик или линкер, в свою очередь, связывают эти перемещаемые адреса с виртуальными адресами.

Каждое связывание — отображение одного адресного пространства в другое.

Связь между файлами программы и утилитами



make — утилита, отслеживающая изменения в файлах и вызывающая необходимые программы из набора, использующегося для компиляции и генерации выполняемого кода из исходных текстов (toolchain).

Привязка инструкций и данных к памяти (настройка адресов) в принципе может быть сделана на следующих шагах:

этап компиляции (Compile time). Когда на стадии компиляции известно точное место размещения процесса в памяти, тогда генерируются абсолютные адреса. Если стартовый адрес программы меняется, необходимо перекомпилировать код. В качестве примера можно привести `.com` программы MS-DOS, которые связывают ее с физическими адресами на стадии компиляции.

этап загрузки (Load time). Если на стадии компиляции не известно где процесс будет размещен в памяти, компилятор генерирует перемещаемый код. В этом случае окончательное связывание откладывается до момента загрузки. Если стартовый адрес меняется, нужно всего лишь перезагрузить код с учетом измененной величины.

этап выполнения (Execution time). Если процесс может быть перемещен во время выполнения из одного сегмента памяти в другой, связывание откладывается до времени выполнения. Здесь желательно специализированное оборудование, например регистры перемещения. Их значение прибавляется к каждому адресу, сгенерированному процессом. Например, x86 использует четыре таких (сегментных) регистра.

Поддержка периферийных устройств и доступ к аппаратуре

Общую структуру вычислительной системы можно представить в следующем виде

Предназначение операционных систем — управлять ресурсами вычислительной системы таким образом, чтобы прикладным программам не приходилось учитывать особенности конкретной аппаратной части.

Устройств, подключаемых к компьютеру, существует огромное множество. Существуют тысячи функционально схожих устройств, но требующих совершенно различных действий для управления ими.

Например, жесткий диск и flash-брелок.

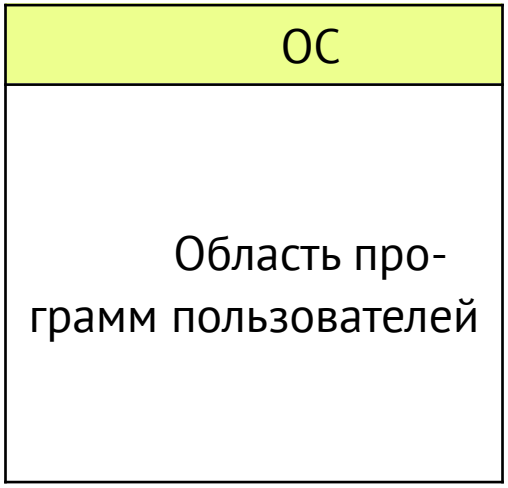
Оба используются для временного хранения файлов и с точки зрения прикладного интерфейса практически не различаются, но по своему строению ничего общего не имеют.

Нестрогая классификация ОС

Мэйнфреймы и малые ЭВМ

1) Пакетные ОС (OS/360/370).

2) Мультипрограммные ОС (MVS) — несколько задач в памяти одновременно. Одна работает, остальные ждут, когда той не понадобится ввод/вывод. Диспетчеризация заданий и управление памятью.



3) Многозадачные ОС (ОС с разделением времени), Time-sharing, Multitasking.

Одновременно обслуживаются несколько пользователей. Используется и диспетчеризация и управление памятью. Появляется понятие процесса.

Интерактивный ввод/вывод (PRIMUS). Виртуальная память.

UNIX (ИНМОС), RT-11 (РАФОС), RSX-11 (ОСРВ).

Процесс — программа, загруженная в память и выполняющаяся.

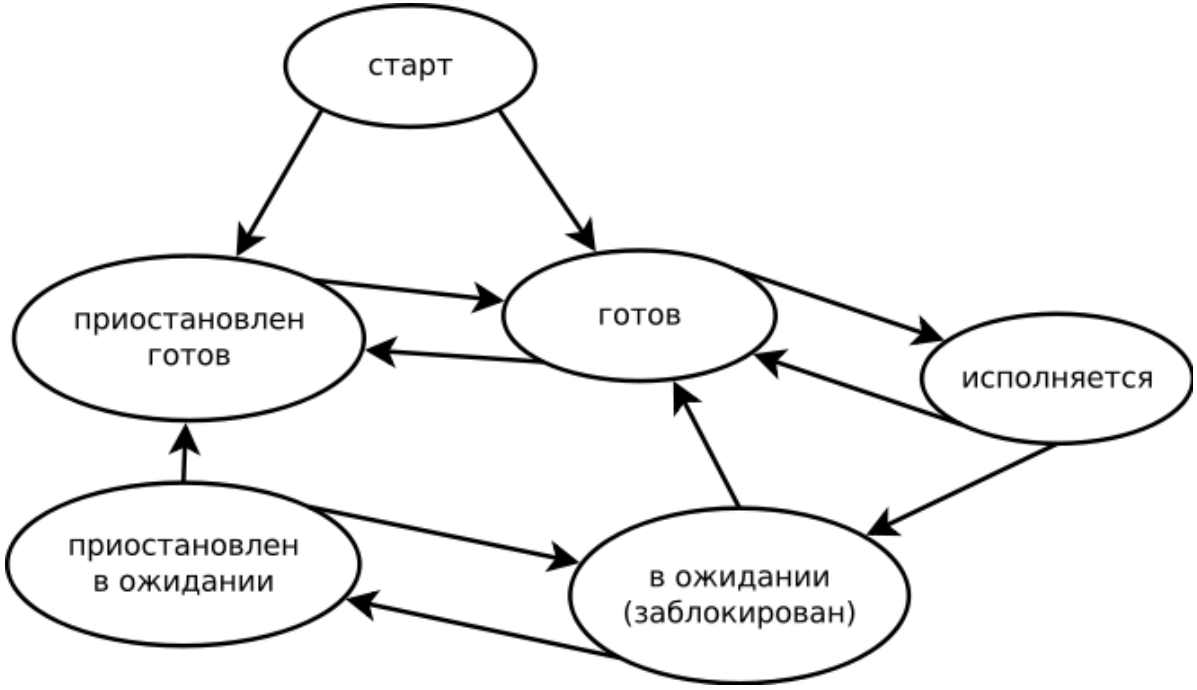
Стандарт ISO 9000:2000 определяет процесс как совокупность взаимосвязанных и взаимодействующих действий, преобразующих входящие данные в исходящие.

Компьютерная программа — только пассивная совокупность инструкций.

Процесс — непосредственное выполнение программных инструкций.

Иногда процессом называют выполняющуюся программу и все её элементы:

- адресное пространство;
- глобальные переменные;
- регистры;
- стек;
- открытые файлы;
- т. д.



Настольные системы

1970-е — микроЭВМ (LSI-11, Электроника-60)
1980-е — IBM PC/XT/AT (CP/M, DR-DOS, IBM PC-DOS, MS-DOS)
Однозадачные операционные системы
Многозадачные — Xenix, IBM OS/2.

Мультипроцессорные системы

Параллельные или тесно связанные системы (SMP).
Все процессоры одинакового ранга (p2p). Память общая.
Иерархические (графический вывод). Собственная память.

Распределенные системы

Коммуникационные возможности.
Сети. Семиуровневый стек протоколов.
LAN — Local-Area Networks — внутри офиса, этажа или здания,
WAN — Wide-Area Networks — между зданиями, городами, странами.
MAN — Metropolitan-Area Networks — между зданиями в городе.

Распределенная система — совокупность независимых компьютеров, которая представляется пользователю единым компьютером.

- сеть рабочих станций (выбор процессора для выполнения программы, единая файловая система);
- роботизированный завод (роботы связаны с разными компьютерами, но действуют как внешние устройства единого компьютера);
- банк со множеством филиалов;
- система резервирования авиабилетов.

Распределенная система представляется пользователям простой системой, в которой он не должен беспокоиться о том, где работают его программы или где расположены файлы, всем этим должна заниматься операционная система.

Системы Клиент-сервер

Клиент-сервер — вычислительная или коммуникационная архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми **серверами**, и потребителями услуг, называемыми **клиентами**.

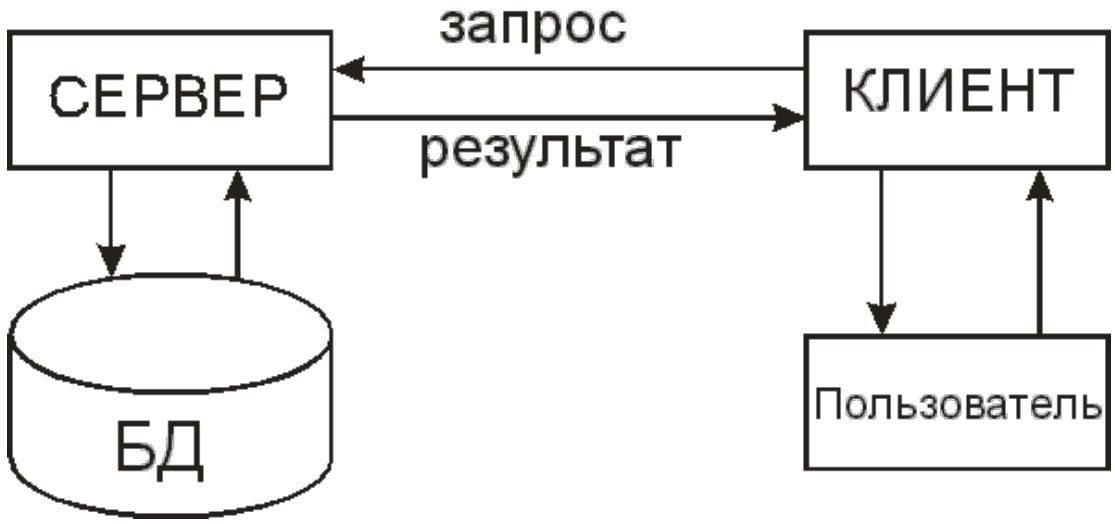
Физически клиент и сервер — это программное обеспечение. Обычно они взаимодействуют через компьютерную сеть посредством сетевых протоколов и находятся на разных вычислительных машинах, но могут выполняться также и на одной машине.

Программы — сервера, ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (файлы, контент, потоковое мультимедиа, работа с базами данных), или сервисных функций.

Сервера вычислений.

Файловые сервера.

Сервера баз данных.



Системы Pear-to-Pear

Слабосвязанные сети PC. ФИДО.
Глобальная сеть INTERNET на основе семейства протоколов TCP/IP.
MS Windows, IBM OS/2, MacOS, UNIX.

Кластерные системы

Высокая доступность.
Несимметричные и симметричные системы.
SAN — Storage-Area network.

ОС реального времени