

Основные понятия и определения

Основы программирования и баз данных

Темы

- Концепции информации
- Основные определения
- Классификация информации
- Алгоритм, его свойства и способы записи
- Программное обеспечение
- Операционная система
- Архитектура и принципы фон Неймана
- Устройство современного компьютера

Концепции информации

- Существование множества определений информации обусловлено сложностью, специфичностью и многообразием подходов к толкованию сущности этого понятия. Существуют 3 наиболее распространенные концепции информации, каждая из которых по-своему объясняет ее сущность.

- **Первая концепция** (концепция К. Шеннона), отражая количественно-информационный подход, **определяет информацию как меру неопределенности (энтропию) события.** Этот подход, хоть и не учитывает смысловую сторону информации, оказался весьма полезным в технике связи и вычислительной технике и послужил основой для измерения информации и оптимального кодирования сообщений.
- При таком понимании **информация - это снятая неопределенность, или результат выбора из набора возможных альтернатив.**

- **Вторая концепция рассматривает информацию как свойство материи.** Ее появление связано с развитием кибернетики и основано на утверждении, что информацию содержат любые сообщения, воспринимаемые человеком или приборами.
- **То есть, информация, как свойство материи, создает представление о ее природе и структуре, упорядоченности и разнообразии. Она не может существовать вне материи, а значит, она существовала и будет существовать вечно, ее можно накапливать, хранить и перерабатывать.**

- **Третья концепция** основана на логико-семантическом подходе, при котором информация трактуется как знание, причем не любое знание, а та его часть, которая используется для ориентировки, для активного действия, для управления и самоуправления.
- **Иными словами, информация - это действующая, полезная часть знаний.**

- Рассмотренные подходы в определенной мере дополняют друг друга, освещают различные стороны сущности понятия информации и облегчают тем самым систематизацию ее основных свойств. Обобщив данные подходы, можно дать следующее определение информации:
- **Информация - это сведения, снимающие неопределенность об окружающем мире, являющиеся объектом хранения, преобразования, передачи и использования. Сведения - это знания, выраженные в сигналах, сообщениях, известиях, уведомлениях и т.д.**

Основные определения

- **Информатика** - (от французского information - информация и automatioque - автоматика) - это область научно-технической деятельности, занимающаяся исследованием процессов получения, передачи, обработки, хранения и представления информации, решением проблем создания, внедрения и использования информационной техники и технологии во всех сферах общественной жизни.

- **Основная задача** информатики заключается в определении общих закономерностей, в соответствии с которыми происходит создание информации, ее преобразование, передача и использование в различных сферах деятельности человека.
- **Прикладные задачи** заключаются в разработке более эффективных методов и средств осуществления информационных процессов, в определении способов оптимальной коммуникации с широким применением технических средств.

- В структуре информатики как науки выделяют **алгоритмическую, программную и техническую** области. Информатика входит в состав кибернетики, изучающей общую теорию управления и передачи информации.
- **Кибернетика** - наука об общих законах получения, хранения, передачи и обработки информации в **сложных системах**. Под сложными системами понимаются технические, биологические и социальные системы. Кибернетика пригодна для исследования любой системы, которая может записывать, накапливать и обрабатывать информацию, благодаря чему ее можно использовать в **целях управления**.

- **Информационная система** - это организованная человеком система сбора, хранения, обработки и выдачи информации, необходимой для эффективного функционирования субъектов и объектов управления. Данные системы являются средством удовлетворения потребностей управления в информации, которое заключается в том, чтобы в нужный момент из соответствующих источников получать информацию, которая должна быть предварительно систематизирована и определенным образом обработана.

- Информационные системы делятся на три класса:
- не производящие качественного изменения информации (учетные, справочные системы);
- анализирующие информацию (аналитические, советующие, прогнозирующие, диагностические системы);
- вырабатывающие решения (управляющие, планирующие системы).

- **Информационные технологии** – это целенаправленный процесс преобразования информации, использующий совокупность средств и методов сбора, обработки, хранения и передачи информации. Как и многие другие технологии, информационная технология должна отвечать следующим требованиям:
 - обеспечивать высокую степень деления всего процесса обработки информации на составляющие компоненты;
 - включать весь набор инструментов, необходимых для достижения поставленной цели;
 - отдельные компоненты должны быть стандартизированы и унифицированы.

- **Информатизация** – внедрение информационных технологий во все сферы человеческой деятельности
- **Инфосфера** – совокупное информационное пространство.

Классификация информации

- Информация подразделяется по форме представления на 2 вида:
- **дискретная форма представления информации;**
- **аналоговая или непрерывная форма представления информации.**

- По способам кодирования выделяют следующие типы информации:
- **символьную**, основанную на использовании символов - букв, цифр, знаков и т. д. Она является наиболее простой, но практически применяется только для передачи несложных сигналов о различных событиях. Примером может служить зеленый свет уличного светофора, который сообщает о возможности начала движения пешеходам или водителям автотранспорта.
- **текстовую**, основанную на использовании комбинаций символов. Здесь так же, как и в предыдущей форме, используются символы: буквы, цифры, математические знаки. Однако информация заложена не только в этих символах, но и в их сочетании, порядке следования. Так, слова КОТ и ТОК имеют одинаковые буквы, но содержат различную информацию. Благодаря взаимосвязи символов и отображению речи человека текстовая информация чрезвычайно удобна и широко используется в деятельности человека: книги, брошюры, журналы, различного рода документы, аудиозаписи кодируются в текстовой форме.
- **графическую**, основанную на использовании произвольного сочетания в пространстве графических примитивов. К этой форме относятся фотографии, схемы, чертежи, рисунки, играющие большое значение в деятельности человек.

- Свойства информации можно рассматривать в трех аспектах:
- **техническом** - это точность, надежность, скорость передачи сигналов и т.д.;
- **семантическом** - это передача смысла текста с помощью кодов;
- **прагматическом** - это насколько эффективно информация влияет на поведение объекта.

Алгоритм и его свойства

- **Алгоритмом** называется точная инструкция исполнителю в понятной для него форме, определяющая процесс достижения поставленной цели на основе имеющихся исходных данных за конечное число шагов.

- Основными свойствами алгоритмов являются:
- Детерминированность
- Массовость
- Результативность
- Дискретность
- Конечность
- Корректность

- **Детерминированность** (определенность, точность, однозначность). Это свойство заключается в том, что при задании одних и тех же исходных данных несколько раз алгоритм будет выполняться абсолютно одинаково, и всегда будет получен один и тот же результат. Свойство детерминированности проявляется также и в том, что на каждом шаге выполнения алгоритма всегда точно известно, что делать дальше, а каждое действие однозначно понятно исполнителю и не может быть истолковано неопределенно. Благодаря этому свойству выполнение алгоритма носит механический характер.

Массовость – выражается в том, что с помощью алгоритма можно решать не одну конкретную задачу, а любую задачу из некоторого класса однотипных задач при всех допустимых значениях исходных данных.

- **Результативность (направленность)** – означает, что выполнение алгоритма обязательно должно привести к решению поставленной задачи, либо к сообщению о том, что при заданных исходных величинах задачу решить невозможно. Алгоритмический процесс не может обрываться безрезультатно.

- **Дискретность** – означает, что алгоритм состоит из последовательности отдельных шагов - элементарных действий, выполнение которых не представляет сложности. Именно благодаря этому свойству алгоритм может быть реализован на ЭВМ.

- **Конечность** — заключается в том, что последовательность элементарных действий алгоритма не может быть бесконечной, неограниченной, хотя может быть очень большой (если требуется, например, большая точность вычислений).

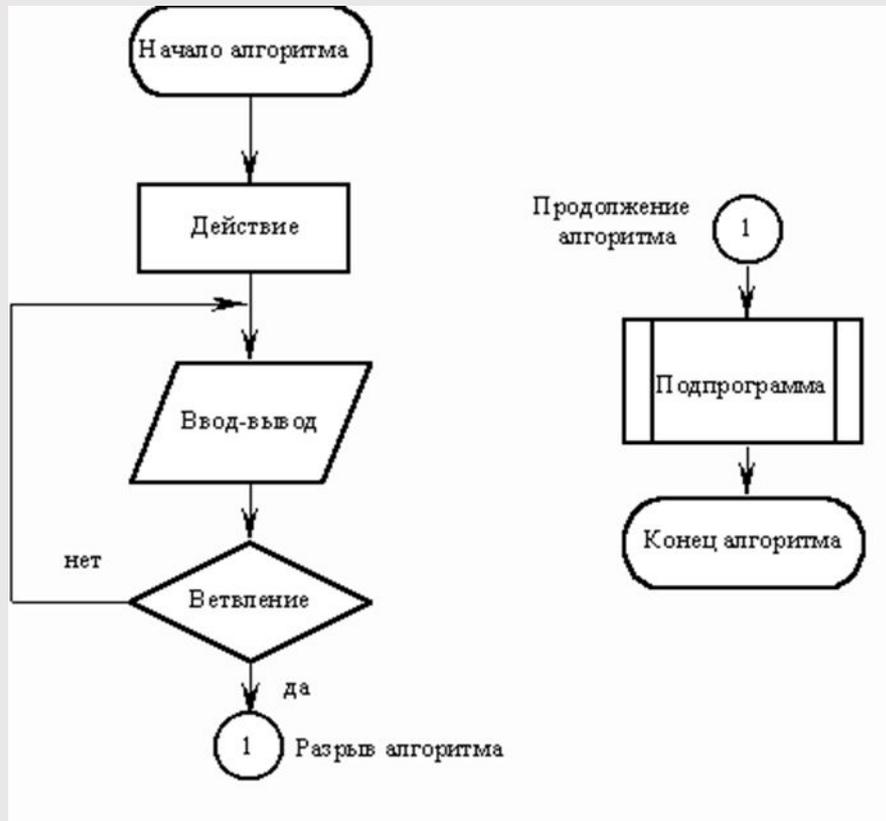
- **Корректность** – означает, что если алгоритм создан для решения определенной задачи, то для всех исходных данных он должен всегда давать правильный результат и ни для каких исходных данных не будет получен неправильный результат. Если хотя бы один из полученных результатов противоречит хотя бы одному из ранее установленных и получивших признание фактов, алгоритм нельзя признать корректным.

- Выделяют три крупных класса алгоритмов:
- **вычислительные алгоритмы**, работающие со сравнительно простыми видами данных, такими как числа и матрицы, хотя сам процесс вычисления может быть долгим и сложным;
- **информационные алгоритмы**, представляющие собой набор сравнительно простых процедур, работающих с большими объемами информации (алгоритмы баз данных);
- **управляющие алгоритмы**, генерирующие различные управляющие воздействия на основе данных, полученных от внешних процессов, которыми алгоритмы управляют.

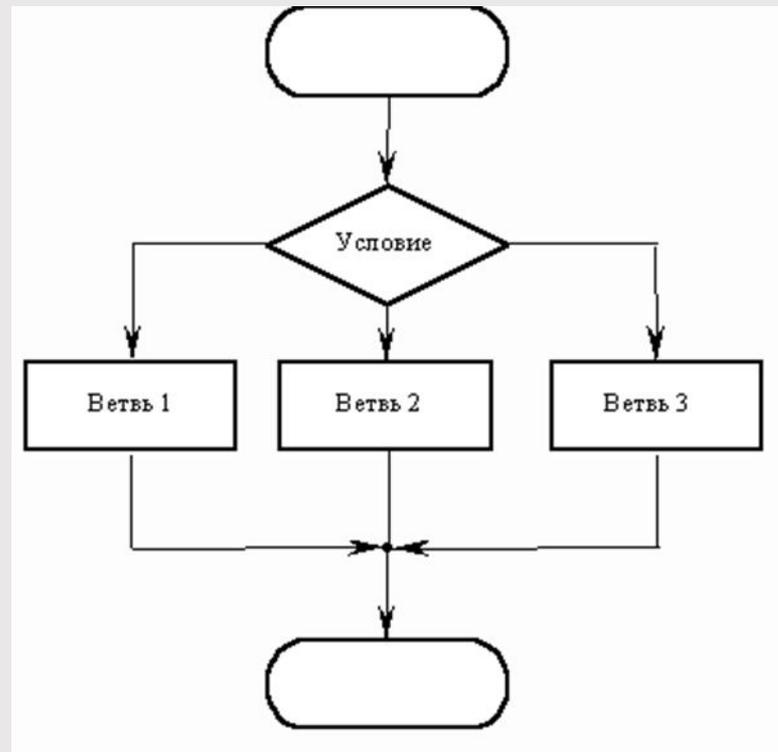
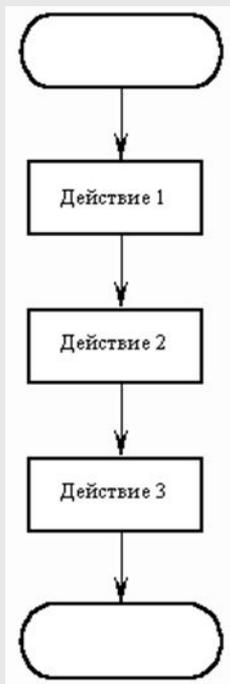
Способы записи алгоритмов

- Для записи алгоритмов используют самые разнообразные средства. Выбор средства определяется типом исполняемого алгоритма. Выделяют следующие основные способы записи алгоритмов:
 - **вербальный**, когда алгоритм описывается на человеческом языке;
 - **символьный**, когда алгоритм описывается с помощью набора символов;
 - **графический**, когда алгоритм описывается с помощью набора графических изображений.

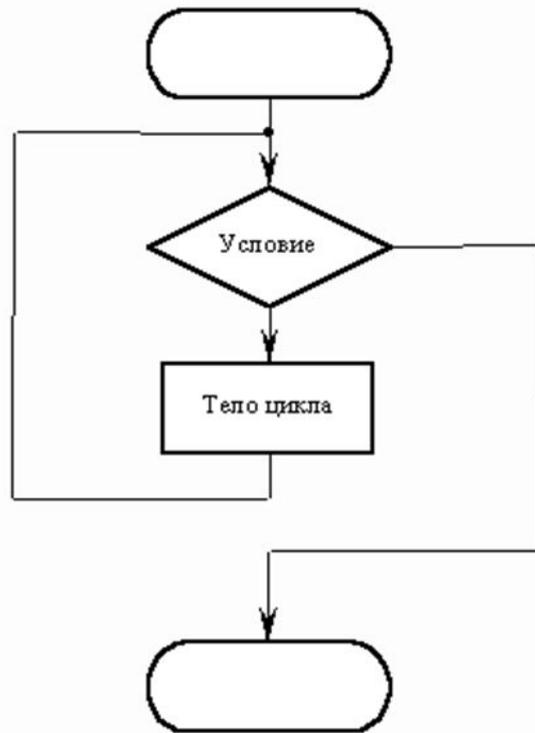
- Описание алгоритма с помощью структурных схем осуществляется рисованием последовательности геометрических фигур, каждая из которых подразумевает выполнение определенного действия алгоритма. Порядок выполнения действий указывается стрелками. Написание алгоритмов с помощью структурных схем регламентируется ГОСТом. Внешний вид основных блоков, применяемых при написании блок схем, приведен на рисунке:



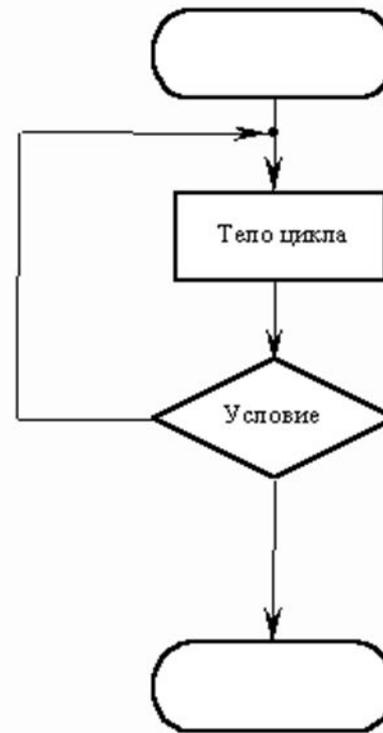
- В зависимости от последовательности выполнения действий в алгоритме выделяют:
- **Алгоритмы линейной структуры**, действия в которых выполняются последовательно одно за другим.
- **Алгоритмы разветвленной структуры**, которые в зависимости от выполнения или невыполнения какого-либо условия производятся различные последовательности действий. Каждая такая последовательность действий называется **ветвью алгоритма**.
- **Алгоритмы циклической структуры**, которые в зависимости от выполнения или невыполнения какого-либо условия выполняется повторяющаяся последовательность действий, называемая **телом цикла**. **Вложенным** называется цикл, находящийся внутри тела другого цикла. Различают циклы с **предусловием** и **послеусловием**.



Цикл с предусловием

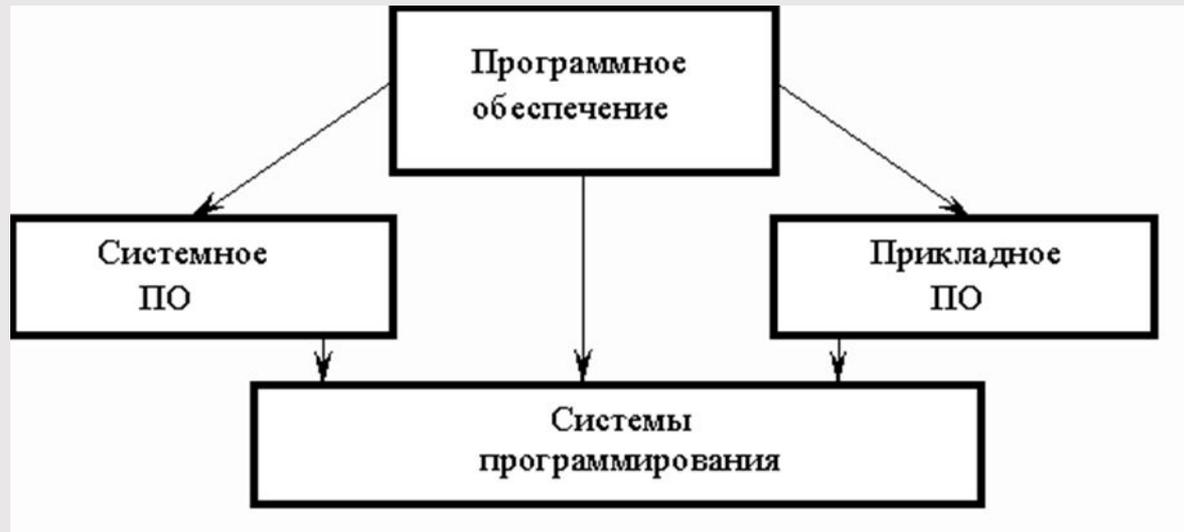


Цикл с послеусловием



Программное обеспечение

- **Программное обеспечение (ПО)** - это совокупность всех программ и соответствующей документации, обеспечивающая использование ЭВМ в интересах каждого ее пользователя.
- Различают системное и прикладное ПО. Схематически программное обеспечение можно представить так:

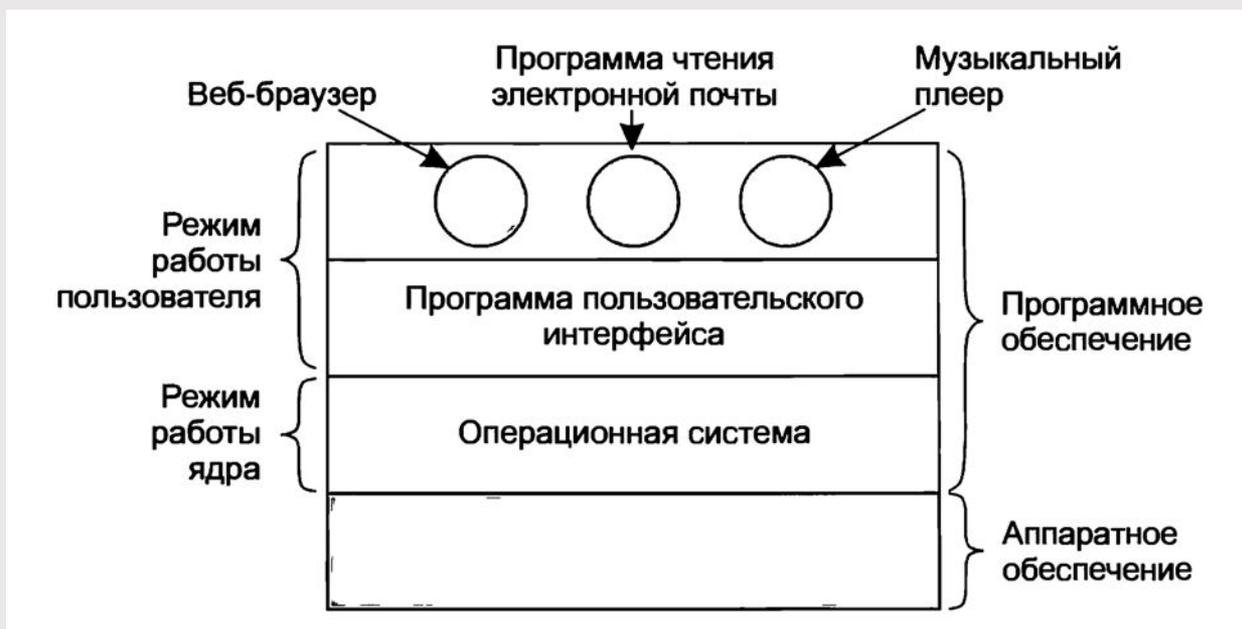


- **Системное ПО** – это совокупность программ для обеспечения работы компьютера. Системное ПО подразделяется на **базовое** и **сервисное**.
- Базовое ПО включает в себя:
 - операционные системы;
 - оболочки.
- Сервисное ПО включает в себя программы (утилиты):
 - диагностики;
 - антивирусные;
 - обслуживания носителей;
 - архивирования;
 - обслуживания сети.

- **Прикладное ПО** – это комплекс программ для решения задач определённого класса конкретной предметной области. Прикладное ПО работает только при наличии системного ПО.
- Прикладные программы называют приложениями. Они включает в себя:
 - текстовые процессоры;
 - табличные процессоры;
 - базы данных;
 - интегрированные пакеты;
 - системы иллюстративной и деловой графики (графические процессоры);
 - экспертные системы;
 - обучающие программы;
 - программы математических расчетов, моделирования и анализа;
 - игры;
 - коммуникационные программы;
 - и т.д.

- Особую группу составляют системы программирования (инструментальные системы), которые являются частью системного ПО, но носят прикладной характер. Системы программирования – это совокупность программ для разработки, отладки и внедрения новых программных продуктов. Системы программирования обычно содержат:
 - трансляторы;
 - компиляторы;
 - среду разработки программ;
 - библиотеки программ (функций, процедур);
 - отладчики;
 - редакторы связей
- и др.

Операционная система



- Большинство процессоров имеет два режима работы: **режим ядра** и **режим пользователя**. Операционная система — наиболее фундаментальная часть программного обеспечения, работающая в режиме ядра (этот режим называют еще режимом супервизора). В этом режиме она имеет полный доступ ко всему аппаратному обеспечению и может задействовать любую инструкцию, которую машина (процессор) в состоянии выполнить. Вся остальная часть программного обеспечения работает в режиме пользователя, в котором доступно лишь подмножество инструкций машины. В частности, программам, работающим в режиме пользователя, запрещено использование инструкций, управляющих машиной или осуществляющих операции ввода-вывода — I/O (Input/Output).

Операционная система как расширенная машина

- Архитектура большинства компьютеров (система команд, организация памяти, ввод-вывод данных и структура шин) на уровне машинного языка машины слишком примитивна и неудобна для использования в программах, особенно это касается систем ввода-вывода.
- **Абстракция** — это ключ к управлению сложностью. Хорошая абстракция превращает практически неподъемную задачу в две, решить которые вполне по силам.
- Первая из этих задач состоит в определении и реализации абстракций. А вторая — в использовании этих абстракций для решения текущей проблемы.
- Задача операционной системы заключается в создании хорошей абстракции, а затем в реализации абстрактных объектов, создаваемых в рамках этой абстракции, и в управлении ими.

- Основные абстракции ОС:
- **Поток** – последовательностью действий процессора во время исполнения программы;
- **Процесс** – исполняемое на компьютере приложение вместе со всеми ресурсами, которые требуются для его исполнения;
- **Адресное пространство** – набора адресов, на которые может ссылаться процесс. Адресное пространство отделено от физической памяти машины и может быть как больше, так и меньше ее по объему;
- **Файловая система** – модель, состоящую из независимых от устройств ввода-вывода файлов и позволяющую скрыть специфику этих устройств.

Операционная система в качестве менеджера ресурсов

- Представление о том, что операционная система главным образом предоставляет абстракции для прикладных программ, — это взгляд сверху вниз. Сторонники альтернативного взгляда, снизу вверх, придерживаются того мнения, что операционная система существует для управления всеми частями сложной системы.
- Управление ресурсами включает в себя мультиплексирование (распределение) ресурсов двумя различными способами: **во времени и в пространстве.**

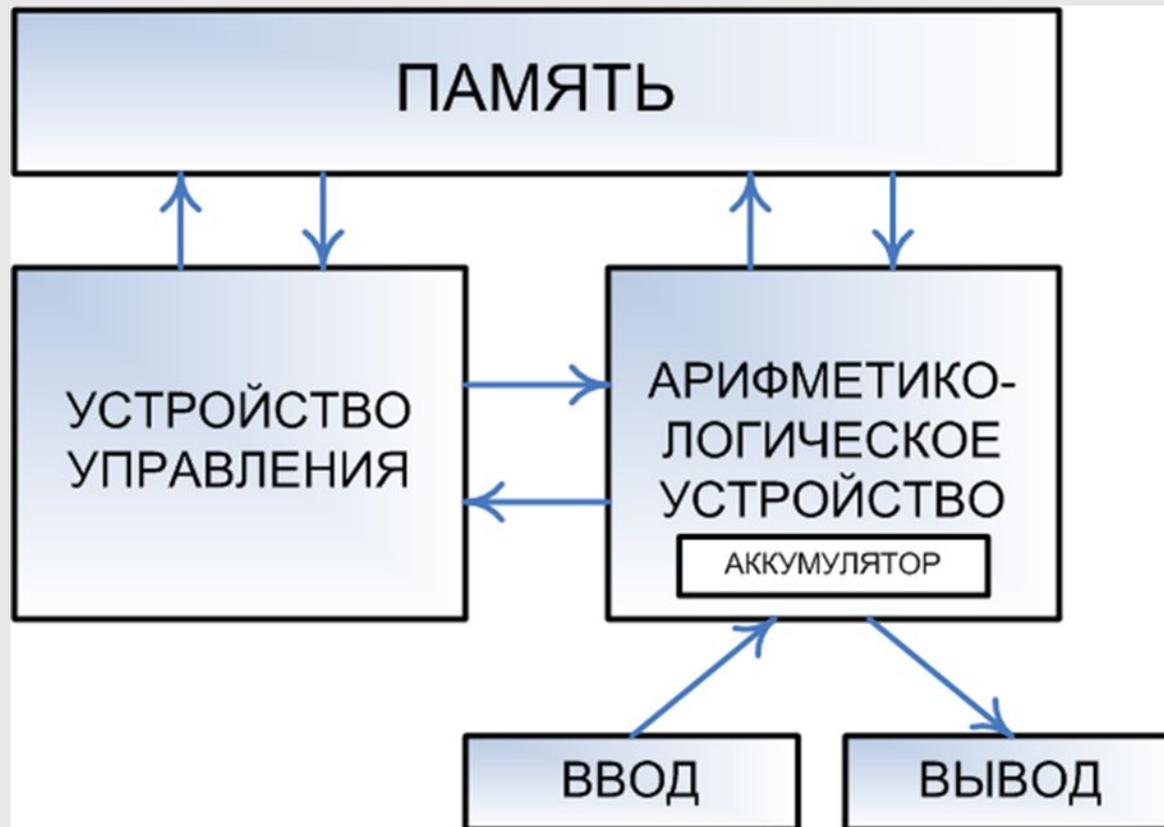
- Когда ресурс разделяется во времени, различные программы или пользователи используют его по очереди. Определение, как именно ресурс будет разделяться во времени — кто будет следующим потребителем и как долго, — это задача операционной системы.
- Другим видом разделения ресурсов является пространственное разделение. Вместо поочередной работы каждый клиент получает какую-то часть разделяемого ресурса. Разумеется, при этом возникают проблемы равной доступности, обеспечения безопасности и т. д., и их должна решать операционная система.

Виды операционных систем

- **Операционные системы мейнфреймов** – ориентированы преимущественно на одновременную обработку множества заданий, большинство из которых требует колоссальных ресурсов и производительности.
- **Серверные операционные системы** – предназначены для серверов, которые представлены очень мощными персональными компьютерами, рабочими станциями или даже универсальными машинами.
- **Многопроцессорные операционные системы** – операционные системы, предназначенные для работы на машинах, объединяющих в себе несколько процессоров.

- **Сетевые операционные системы** – операционные системы поддерживающие работу в сети, в таких системах пользователи знают о существовании множества компьютеров и могут войти в систему удаленной машины.
- **Распределенные операционные системы** – операционные системы, представляющиеся своим пользователям как традиционная ОС, хотя на самом деле в их составе может работать множество компьютеров.
- **Операционные системы персональных компьютеров** – операционные системы, основной задачей которых является качественная поддержка работы отдельного пользователя.
- **Операционные системы реального времени** – операционные системы, предназначенные для управления технологическими процессами.

Принципы и архитектура фон Неймана



- **Принцип однородности памяти**

Принципиальное отличие фон-неймановской (принстонской) архитектуры от **гарвардской**. Команды и данные хранятся в одной и той же памяти и внешне в памяти неразличимы. Распознать их можно только по способу использования; то есть одно и то же значение в ячейке памяти может использоваться и как данные, и как команда, и как адрес в зависимости лишь от способа обращения к нему. Это позволяет производить над командами те же операции, что и над данными, и, соответственно, открывает ряд возможностей. Так, циклически изменяя адресную часть команды, можно обеспечить обращение к последовательным элементам массива данных. Такой прием носит название **модификации команд** и с позиций современного программирования не приветствуется. Более полезным является другое следствие принципа однородности, когда команды одной программы могут быть получены как результат исполнения другой программы. Эта возможность лежит в основе **трансляции** — перевода текста программы с языка высокого уровня на язык конкретной вычислительной машины.

- **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, должны быть представлены в виде программы, состоящей из последовательности управляющих слов — команд. Каждая команда предписывает некоторую операцию из набора операций, реализуемых вычислительной машиной. 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. Последовательность битов в формате, имеющая определенный смысл, называется полем. В числовой информации обычно выделяют поле знака и поле значащих разрядов. В формате команды в простейшем случае можно выделить два поля: поле кода операции и поле адресов.

Устройство компьютера

- Концептуально простой персональный компьютер можно представить в виде следующей модели: центральный процессор, память и устройства ввода-вывода соединены системной шиной, по которой они обмениваются информацией друг с другом. Современные персональные компьютеры имеют более сложную структуру и используют несколько шин, но для начала вполне достаточно и этой модели.

Процессор

- Центральный процессор (ЦП) — это «мозг» компьютера. Он выбирает команды из памяти и выполняет их. Обычный цикл работы центрального процессора выглядит так: **выборка** из памяти первой команды, ее **декодирование** для определения ее типа и операндов, **выполнение** этой команды, а затем выборка, декодирование и выполнение последующих команд. Этот цикл повторяется до тех пор, пока не закончится программа. Таким образом, осуществляется выполнение программ.
- Поскольку доступ к памяти для получения команды или данных занимает намного больше времени, чем выполнение команды, у всех центральных процессоров есть несколько собственных **регистров** для хранения основных переменных и промежуточных результатов.

- Для повышения производительности процессоров их разработчики давно отказались от простой модели извлечения, декодирования и выполнения одной команды за один цикл. Многие современные процессоры способны одновременно выполнять более одного действия. Например, у процессора могут быть **отдельные блоки для выборки, декодирования и выполнения команд**, тогда во время выполнения команды n он сможет декодировать команду $n + 1$ и осуществлять выборку команды $n + 2$. Подобная организация работы называется **конвейером**.

- Более совершенной конструкцией, по сравнению с конвейерной, обладает **суперскалярный процессор**. Он имеет **несколько исполнительных блоков**: например, один — для целочисленной арифметики, другой — для арифметики чисел с плавающей точкой, а третий — для логических операций.

- После суперскалярной архитектуры, следующим очевидным шагом является **дублирование** не только функциональных блоков, но и части **управляющей логики**. Такие свойства, присущие процессору называются **многопоточностью**, или **гиперпоточностью** (**hyperthreading** по версии Intel). В первом приближении эта технология позволяет процессору сохранять состояние двух различных потоков и осуществлять переключения между ними за наносекунды.

- Кроме процессоров с многопоточностью, существуют процессоры, содержащие на одном кристалле два, четыре и более полноценных процессора, или ядра. Несомненно, для использования такого **многоядерного процессора** потребуется многопроцессорная операционная система.

Память

- Второй основной составляющей любого компьютера является память. В идеале память должна быть максимально быстрой (работать быстрее, чем производится выполнение одной инструкции, чтобы работа центрального процессора не замедлялась обращениями к памяти), достаточно большой и чрезвычайно дешевой.
- Никакая современная технология не в состоянии удовлетворить все эти требования, поэтому используется другой подход. Система памяти создается в виде иерархии уровней. Верхние уровни обладают более высоким быстродействием, меньшим объемом и более высокой удельной стоимостью хранения одного бита информации, чем нижние уровни, иногда в миллиарды и более раз.



Единицы измерения количества информации

- Бит — это базовая единица измерения количества информации, равная количеству информации, содержащемуся в опыте, имеющем два равновероятных исхода.
- Байт – единица хранения и обработки цифровой информации; совокупность битов. В современных вычислительных системах байт состоит из восьми битов.
- Приставки:
 - К – 1024;
 - М – 1024К;
 - Г – 1024М;
 - и т.д.

Устройства ввода-вывода

- Устройства ввода-вывода обычно состоят из двух компонентов: **физического устройства** и его **контроллера**.
- Контроллер представляет собой микросхему или набор микросхем, которые управляют устройством на физическом уровне. Он принимает от операционной системы команды, а затем их выполняет.
- Программа, обеспечивающая взаимодействие ОС с контроллером, выдачи ему команды и воспринятая поступающих от него ответов, называется драйвером устройства.

- В каждом контроллере для связи с ним имеется небольшое количество **регистров**.
- На некоторых компьютерах регистры устройств отображаются в адресное пространство операционной системы (на те адреса, которые она может использовать), поэтому состояния регистров можно считывать и записывать точно так же, как и обычные слова в оперативной памяти.
- На других компьютерах регистры устройств помещаются в специальное пространство портов ввода-вывода (I/O port space), в котором каждый регистр имеет адрес порта. На таких машинах в режиме ядра доступны специальные команды ввода-вывода (обычно обозначаемые IN и OUT), позволяющие драйверам осуществлять чтение и запись данных в регистры.

- Ввод и вывод данных можно осуществлять тремя различными способами:
- Активное ожидание (постоянный опрос устройства).
- По прерыванию (по завершению операции устройство вырабатывает прерывание).
- Использование контроллера прямого доступа к памяти (DMA). Работой с устройством занят не процессор, а специальный контроллер.

Выбирайте Центр «Специалист» – крупнейший учебный центр России!

info@specialist.

RU (495)

232-32-16