

# **ВВЕДЕНИЕ**

# Оперативная память

Память компьютера лучше представить себе в виде последовательности ячеек.

Каждая ячейка - один байт.

Байты (ячейки) памяти пронумерованы один за другим, начиная с 0.

Любая информация в памяти может занимать один или несколько байтов.

# **Из чего состоит программа?**

Программа состоит из данных и команд, которые работают с этими данными.

# Как работает программа

При запуске программы, она с жесткого диска загружается на свободное место в оперативную память. При этом программа:

1. получает входные данные с устройств ввода (клавиатура, мышка) и заносит их в оперативную память;
2. Обрабатывает введенные данные с помощью команд;
3. Результат обработки (выходные данные) из оперативной памяти выводит на устройства вывода (монитор, жесткий диск, принтер).

# Что размещается в оперативной памяти

В оперативной памяти размещаются:

- входные данные (то, что дано)
- выходные данные (результат работы программы)
- промежуточные данные (если они нужны)
- команды, с помощью которых из входных данных получаются выходные.

# Что такое данные

Данные это – информация необходимая для работы команд, а также результат выполнения команды.

Пока мы будем рассматривать только числовые данные.

Числа могут быть:

- Целыми и вещественными (с дробной частью).
- Очень большими и очень маленькими.
- Положительными и отрицательными.

Поэтому существуют различные типы данных.

# Что определяет тип данных?

- Сколько байтов оперативной памяти мы будем отводить этому данному (насколько большое или маленькое число)
- Как нужно интерпретировать эти данные
  - Целое число (со знаком или без знака)
  - Или число вещественное (с дробной частью)
  - Или это символ

# Что такое команда

Команда – это операция, которую компьютер должен выполнить с данными. Например, нужно вычислить сумму чисел 2 и 3. Что нужно, чтобы записать команду?

- Код (условное обозначение, в данном случае +)
- Входные данные (операнды, в данном случае числа 2 и 3)
- Результат (тоже данное, в данном случае 5)

# Зачем данным нужны имена

Программа начинает выполняться начиная с первой команды и выполняет ПОДРЯД все команды.

Но команды работают с данными. Как же команде обратиться к этим данным?

Они расположены где-то в оперативной памяти, но мы не знаем по каким адресам.

Поэтому в команде вместо адресов, по которым расположены данные, указываются имена.

# Что такое имя (идентификатор)?

Ячейка в оперативной памяти, которая имеет имя называется переменной или константой.

Указывая в программе имя, мы фактически указывает адрес в оперативной памяти.

# Что такое значение?

То число, которой находится в ячейке оперативной памяти, по этому адресу (под этим именем), называется значением переменной или константы.

Значение переменной можно менять, значение константы менять нельзя.

# Как написать программу

Для того, чтобы компьютер вас правильно понял, нужно написать программу. Программа должна быть написана на каком-то языке программирования.

Поэтому нужно на этом языке грамотно записать последовательность команд, с помощью которых из входных данных получаются выходные.

# Виды языков программирования

Все языки программирования делятся на два вида — языки низкого и высокого уровня.

**Языки низкого уровня** — программа записывается на языке, близком к машинному.

Такие языки жестко привязаны к определенному типу аппаратуры (у каждого типа процессора — свой машинный код). Пример?

Пример языка низкого уровня – ассемблер

# Виды языков программирования

**Языки высокого уровня** – это языки программирования, позволяющие записывать программу в удобном для человека виде.

Такие языки не привязаны к типу процессора и значительно проще в использовании.

Таких языков много. Мы будем заниматься C++, C#, JavaScript, SQL

# Что из себя представляет программа

Физически программа на языке высокого уровня (C++) представляет собой текстовый файл (с расширением `cpp`), в котором представлены данные и команды данного языка в заданном программистом порядке.

Программа может также содержать обращение к уже написанным функциям (подпрограммам), которые хранятся в отдельных библиотечных файлах.

# Для чего нужен транслятор?

Чтобы компьютер мог понять программу, которая написана на языке высокого уровня, нужно ее перекодировать — перевести на машинный язык.

Это выполняется специальными программами, которые называются трансляторами.

# Транслятор-компилятор, что это?

Трансляторы бывают двух видов: компиляторы и интерпретаторы.

Компилятор читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Для выполнения законченного варианта программы в таких языках исходный текст программы не нужен.

Компиляторы у таких языков: Pascal, C, C++

# Транслятор-интерпретатор, что это?

Интерпретатор переводит и выполняет программу строка за строкой.

В отличие от компилятора, интерпретатор НЕ порождает на выходе программу на машинном языке. Распознав команду исходного языка, он тут же выполняет ее.

Поэтому исходный текст программы нужен для ее выполнения.

Интерпретаторы: Java Script, Python

# Сборка, зачем нужна?

Компилятор создает не готовую к исполнению программу, а только объектный код (файл с расширением \*.obj).

Этот код содержит результат перевода на машинный язык текста программы, созданной программистом.

Но создаваемая программа может содержать обращения к функциям стандартных библиотек, реализации которых описаны в отдельных файлах.

# Сборка

Для объединения кода создаваемой программы и кода библиотечных функций используется специальная программа.

В результате сборки получается исполнимый файл \*.exe

Все эти действия выполняются в специальной среде программирования.

# Среда программирования

Среда программирования объединяет в себе:

- Текстовый редактор для написания текста программы.  
(В результате получается текстовый файл \*.crr или другим расширением, в зависимости от языка)
- Транслятор (В результате получается файл \*.obj)
- Компоновщик (В результате получается файл \*.exe)

# Что такое проект

Проект это совокупность файлов и папок объединенных в единое целое для решения определенной задачи.

Как правило, в проект входят:

- Текстовый файл (\*.crr), содержащий текст программы;
- Файл \*.obj, содержащий оттранслированную программу;
- Файл \*.exe, содержащий готовую к выполнению программу, с уже подключенными библиотечными функциями;
- Могут быть другие файлы, содержащие различные настройки.

# Технологии программирования

Технологией программирования называют совокупность средств, используемых в процессе разработки текста программы.

Языки высокого уровня делятся на:

- процедурные;
- логические;
- объектно-ориентированные.

# Процедурное программирование

Процедурные языки предназначены для однозначного описания алгоритмов.

При решении задачи процедурные языки требуют в той или иной форме явно записать процедуру ее решения.

Наиболее известные процедурные языки: PL/1, Pascal, C

# Логическое программирование

Логические языки ориентированы не на запись алгоритма решения задачи, а на формализованное описание условия задачи.

В этих языках указывается что дано и что требуется получить. При этом поиск решения задачи возлагается непосредственно на ПК.

Самым известным языком логического программирования является Prolog.

# Объектно-ориентированное программирование

Основная идея ООП заключается в

объединении данных с методами, которые

эти данные обрабатывают в единое целое —  
объект.

Объектно-ориентированные языки: C++,  
Java, C#