

Sem 1.01.1 Об устройстве ЭВМ

Презентация к семинару в рамках курса
«Программирование на Си»
Подготовил Кострицкий А. С.

Кафедра ИУ-7
МГТУ имени Н. Э. Баумана
Москва, Россия

Москва — 2023 — TS2302162026

Устройство современного персонального компьютера слишком сложно, чтобы можно было подробно его рассматривать в рамках нашего курса.

На протяжении всего года будем считать, что мы пишем код под некую общую модель компьютера, о свойствах которой мы сейчас поговорим.

Все наши компьютеры на самом деле очень похожи. Вы можете открыть инструкцию или гарантийный талон собственной машины и, в большинстве случаев, увидеть там официальное название «IBM-PC-совместимая ПЭВМ» (англ. *IBM PC compatible computer*). Будем причислять наш компьютер к потомкам *принстонской архитектуры*, иначе говоря, *архитектуры фон Неймана*. Причём не столько потому, что наши ПК такие уж «принстонские», а потому, что к любому другому семейству отнести наши ПК ещё сложнее.

Будем считать, что внутри нашего компьютера есть три составляющих.

Первая часть, *центральный процессор* (англ. *central processor unit*), выполняет несколько функций.

Находящееся внутри процессора АЛУ работает как калькулятор для данных, попадающих в процессор.

УУ управляет загрузкой/выгрузкой данных памяти и оборудованием.

Внутри ЭВМ думает только процессор. Ни один приказ не может пройти мимо процессора.

Вторая составляющая, *оперативное запоминающее устройство, ОЗУ*, предоставляет место для хранения данных **и** кода. ОЗУ будем представлять в виде одномерного массива ячеек, к каждой из которых процессор может обратиться по *адресу*.

На английском этот термин звучит не иначе, как *random access memory, RAM*, причём слово «random» стоит переводить как «произвольный».

Определение

Минимальную разницу между двумя разными адресами мы будем называть **минимальной единицей адресации**.

В нашей машине положим её равной одному байту.

Третья составляющая, система ввода-вывода (IO), позволяет процессору общаться с внешним оборудованием, включая оборудование для взаимодействия с пользователем.

Стоит отметить, что жёсткий диск мы тоже будем считать внешним устройством, с которым процессор общается через IO.

Для упрощения абстракции будем выделять несколько характерных величин. Про первую, минимальную единицу адресации, мы уже сказали.

Очевидно, что процессор не может общаться с другими частями ПК с помощью сообщений в бит — накладные расходы такого общения были бы больше полезной информации.

Определение

*Будем считать, что составные части компьютера общаются сообщениями фиксированной длины, **машинными словами**.*

Будем полагать, что величину машинного слова для нашей машины мы не знаем, но на практике будем ограничиваться выбором между 4 и 8 байтами.

Будем разделять свойства программы на три большие группы. В первую будем относить всё, что касается *скорости работы*, во вторую — всё относящееся к *объёму запрашиваемой памяти*, а в третью — всё, что соотносится с *поддерживаемостью*, которую можно отождествлять с *читабельностью* и *модульностью*.

Будем считать, что при прочих равных для нас наиболее важна поддерживаемость программы. *Программы пишутся людьми для людей, а преждевременная оптимизация — корень всех проблем.* Помимо этого, будем считать, что скорость важнее памяти.

Вопросы для самопроверки I

- 1 Можете ли Вы назвать какие-либо компоненты ЭВМ, которые не вошли в модель? Почему Вы решили, что они именно не присутствуют в модели, а не просто учитываются в одном из названных компонентов неявно?
- 2 Почему жёсткий диск мы считаем внешним оборудованием? Есть ли у жёсткого диска своё управляющее устройство?
- 3 Сколько памяти можно адресовать, если на компьютере фиксирована величина адреса четырьмя байтами? Восемью? Шестнадцатью? n байтами?
- 4 Как Вы думаете, чем отличаются x32 и x64 архитектуры?

Вопросы для самопроверки II

- 5 Можете ли Вы назвать примеры машин, не относящихся к принстонской архитектуре?
- 6 Какую классификацию Вы можете предложить для устройств ввода-вывода?