**Тема 5. Управление памятью**

**Лекция 16 «Структура оперативной памяти. Адресация. Основные регистры».**

**Оперативная память** - это память, в которой во время работы ПК хранятся программы и данные. Файлы программы при ее запуске загружаются в **оперативную память**, в которой хранятся во время работы с указанной программой. Процессор выполняет программно-реализованные команды, содержащиеся в памяти и сохраняет их результаты.

Во время выполнения программы в оперативной памяти хранятся ее данные. Оперативная память хранит коды нажатых клавиш при работе с текстовым редактором, а также величины математических операций. Микросхемы оперативной памяти (RAM) называют энергозависимой памятью: после выключения компьютера данные, хранимые в них, будут «потеряны№, если они предварительно не были сохранены. При выполнении команды Сохранить (Save) содержимое оперативной памяти сохраняется в виде файла на жестком диске или другом устройстве внешней памяти.

Устройства оперативной памяти иногда называют запоминающими устройствами с произвольным доступом. Это означает, что обращение к данным, хранящимся в оперативной памяти, не зависит от порядка их расположения в ней.

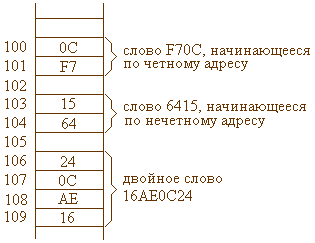
Физически оперативная память в системе представляет собой набор микросхем или модулей, содержащих микросхемы, которые обычно подключаются к системной плате. Эти микросхемы или модули могут иметь различные характеристики и, чтобы функционировать правильно, должны быть совместимы с системой, в которую устанавливаются.

В современных ПК используются **запоминающие устройства трех основных типов.**

* **DRAM (Dynamic Random Access Memory) -** динамическое запоминающее устройство с произвольным доступом.
* **SRAM (Static RAM)**. Статическая оперативная память.
* **ROM (Read Only Memory)**. Постоянное запоминающее устройство — ПЗУ, не способное выполнять операцию записи данных.

**Структура оперативной памяти.**

Оперативная память состоит из отдельных байт, причем каждому байту назначен уникальный адрес (беззнаковое число). Адреса начинаются с 0, а последний доступный адрес определяется моделью используемого процессора и режимом его работы. Например, для процессора 8086, последним доступным адресом будет FFFFF. Любые два смежных байта образуют слово. За адрес слова принимается адрес его младшего байта. Для 32 разрядных моделей, любые четыре смежных байта образуют двойное слово. За адрес двойного слова принимается адрес его младшего слова, а, следовательно, адрес его младшего байта. Байты, слова и двойные слова могут располагаться с любого адреса. Однако для слов и двойных слов существует понятие выравнивания. Операции со словами, например, будут выполняться быстрее, если слово начинается с адреса, кратного 2, или, по-другому, с четного адреса. В некоторых случаях выравнивание обязательно. Иногда требуется выравнивание по адресу кратному 4, 8, 16, или даже 4096.



Кроме того, с точки зрения программиста, память разделена на сегменты. Это вызвано тем, что процессор 8086 оперирует 16-разрядными числами, следовательно, адресуемая память составляет 216=65536 байт (с 0 до 65535), чего явно недостаточно. Поэтому адресная шина процессора состоит из 20 разрядов (для 8086), что позволяет адресовать 1М байт. Сегментная организация и позволяет отобразить 16-разрядный адрес процессора в 20-разрядный (для 8086) адрес памяти. Более того, такая организация позволяет легко писать программы, которые можно перемещать по памяти. При этом требуется только поменять содержимое сегментного регистра, а саму программу можно оставить неизмененной. Это называется позиционная независимость. Современные модели процессоров работают с 32-разрядными, или 64-разрядными адресами. Но для совместимости сверху вниз, с процессором 8086, сегментная схема адресации в них сохранена.

**Адресация.**

**Адресация -** осуществление ссылки (обращение) к устройству или элементу данных по его адресу; установление соответствия между множеством однотипных объектов и множеством их адресов; метод идентификации местоположения объекта.

**Адрес памяти** представляет собой последовательность цифр фиксированной длины, которые, обычно, отображаются и обрабатываются как целые числа. Причиной такой формы записи являются особенности [CPU](https://ru.bmstu.wiki/CPU_(Central_Processing_Unit)) (такие как программный счетчик и возрастающие регистры адреса(ячейки) памяти), а также использование памяти как массива, поддерживаемого различными языками программирования.

**Методы адресации:**

1. **Адресное пространство:**

* Простая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) flat addressing) — указание объекта с помощью идентификатора или числа, не имеющего внутренней структуры.
* Расширенная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) extended addressing) — доступ к запоминающему устройству с адресным пространством, большим диапазона адресов, предусмотренного форматом [команды](https://dic.academic.ru/dic.nsf/ruwiki/641375).
* Виртуальная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) virtual addressing) — принцип, при котором каждая программа рассматривается как ограниченное непрерывное поле логической памяти, а адреса этого поля — как [виртуальные адреса](https://dic.academic.ru/dic.nsf/ruwiki/1720104).
* Ассоциативная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) associative addressing) — точное местоположение данных не указывается, а задаётся значение определённого поля данных, идентифицирующее эти данные.

1. **Исполнение программ:**

* Статическая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) static addressing) — соответствие между [виртуальными](https://dic.academic.ru/dic.nsf/ruwiki/1720104) и [физическими адресами](https://dic.academic.ru/dic.nsf/ruwiki/1720104) устанавливается до начала и не меняется в ходе выполнения программы.
* Динамическая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) dynamic addressing) — преобразование виртуальных адресов в физические осуществляется в процессе выполнения программы. Программа при этом не зависит от места размещения в физической памяти и может перемещаться в ней в процессе выполнения.

1. **Кодирование адресов:**

* Явная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) explicit addressing) — адресация путём явного задания адресов в программе.
* Неявная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) implied addressing) — один или несколько [операндов](https://dic.academic.ru/dic.nsf/ruwiki/82036) или [адресов операндов](https://dic.academic.ru/dic.nsf/ruwiki/1720104) находятся в фиксированных для данной [команды](https://dic.academic.ru/dic.nsf/ruwiki/641375) [регистрах](https://dic.academic.ru/dic.nsf/ruwiki/179302) или ячейках памяти и не требуют явного указания в команде.
* Абсолютная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) absolute addressing) — адресная часть команды содержит [абсолютный адрес](https://dic.academic.ru/dic.nsf/ruwiki/1720104).
* Символическая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) symbolic addressing) — адресная часть команды содержит [символический адрес](https://dic.academic.ru/dic.nsf/ruwiki/1720104).

1. **Вычисление адресов:**

* Непосредственная, прямая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) immediate (direct) addressing) — адресная часть команды содержит [непосредственный (прямой) адрес](https://dic.academic.ru/dic.nsf/ruwiki/1720104); адресация путём указания прямых адресов.
* Косвенная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) indirect addressing) — адресная часть команды содержит [косвенный адрес](https://dic.academic.ru/dic.nsf/ruwiki/1720104); адресация посредством косвенных адресов.
* Регистровая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) register addressing) — задание адресов операндов в [регистрах](https://dic.academic.ru/dic.nsf/ruwiki/179302).
* Базисная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) basic addressing) — вычисление адресов в машинных командах относительно содержимого регистра, указанного в качестве базового.
* Базовая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) base-displacement addressing) — схема вычисления [исполнительного адреса](https://dic.academic.ru/dic.nsf/ruwiki/1720104), при которой этот адрес является суммой [базового адреса](https://dic.academic.ru/dic.nsf/ruwiki/1720104) и [смещения](https://dic.academic.ru/dic.nsf/ruwiki/1720104).
* Относительная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) relative addressing) — адресная часть команды содержит [относительный адрес](https://dic.academic.ru/dic.nsf/ruwiki/1720104).
* Индексная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) indexed addressing) — формирование исполнительного адреса осуществляется путём добавления к базовому адресу содержимого [индексного регистра](https://dic.academic.ru/dic.nsf/ruwiki/1378194).
  + Автодекрементная, автоинкрементная ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) autodecremental, autoincremental addressing — содержимое регистра индекса изменяется (уменьшается или увеличивается) на некоторое число.
  + Постдекрементная, предекрементная, постинкрементная, преинкрементная — автодекрементные и автоинкрементные адресации, при которых уменьшение/увеличение происходит после/до выборки операнда.
* Стековая ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) stack addressing) — адресация посредством регистра — указателя [стека](https://dic.academic.ru/dic.nsf/ruwiki/8488).
* Самоопределяющаяся ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) self-relative addressing) — адресная часть команды содержит [самоопределяющийся адрес](https://dic.academic.ru/dic.nsf/ruwiki/1720104).
  + Адресация относительно счётчика команд ([англ.](https://dic.academic.ru/dic.nsf/ruwiki/6161) program counter relative addressing) — адреса в команде указываются в виде разности исполнительных адресов и адреса исполняемой команды. Такой способ адресации не требует настройки.

**Основные регистры.**

Кроме оперативной памяти существует память, расположенная непосредственно в процессоре – это регистровая память. Отдельные ячейки памяти называются регистрам, регистры не имеют собственных адресов, для доступа к ним используются специальные форматы машинных команд.

Основным отличием регистра от ячейки памяти, кроме расположения является использование его для хранения управляющей информации, а не произвольных данных (единственное исключение составляют регистры общего назначения). Регистровая память имеется не только в процессоре, но и периферийных устройствах и контроллерах. Ими ведь тоже надо управлять. Причем контроллеры периферийных устройств могут иметь не только регистры, но и собственную память. Например, контроллер дисплея имеет и регистры (регистр режима, регистр битовой маски и т.д.) и оперативную память, называемую видеопамятью.

Процессоры Intel 80х86 имеют несколько групп регистров:

* **Регистры общего назначения.** AL, AH, AX, EAX, BL, BH, BX, EBX, CL, CH, CX, ECX, DL, DH, DX, EDX. Эти регистры используются для промежуточного хранения результатов операций. Некоторые регистры общего назначения имеют и специальное назначение. Например, регистры CL, CX и ECX используются как счетчики в командах сдвигов.
* **Указательные и индексные регистры.** SP - указатель стека, BP - указатель базы, SI - индекс источника, DI - индекс приемника. Эти регистры могут использоваться как явным, так и не явным образом. Например, регистр SI можно использовать и как регистр общего назначения, для хранения промежуточных результатов, и как специализированный регистр, для последовательного доступа к символам в символьной строке. Регистр SP автоматически используется в операциях со стеком, в том числе при вызове процедур.
* **Сегментные регистры.** CS - сегмент кода программы, DS - сегмент данных, SS - сегмент стека, ES - сегмент дополнительных данных. Начиная с процессора 80386 добавились еще два сегментных регистра, FS и GS. Их назначение такое же, как и регистра ES. Эти регистры используются для хранения сегментной части логического адреса.
* **Указатель команды (счетчик команд).** Регистр IP всегда содержит адрес следующей выполняемой машинной команды.
* **Регистр флажков и слова состояния процессора.** Этот регистр называют по-разному: PSW - слово состояния процессора, FLAGS - регистр флагов, EFLAGS - расширенный регистр флагов. Суть от названия не меняется. Он содержит признаки результата и другую информацию о состоянии процессора. Кстати, в программах на Ассемблере его **всегда** называют PSW.
* **Регистры специального назначения, системные и управляющие регистры.** Как следует из названия, в эту группу входят регистры, используемые, в основном, системными программистами и операционными системами. Они определяют расположение в оперативной памяти системных таблиц, режимов работы процессора, функционирование механизмов защиты, содержат дополнительную информацию о текущем состоянии процессора.