**Адресация в реальном режиме**

* **Сегмент** – участок памяти размером 64 КБ (начиная с физ. адреса: 16 бит)
* **Смещение** – порядок номера байта относительно начало сегмент
  + Начиная с 0
* Полный адрес состоит из:
  + Номера сегмента (16 бит)
  + Смещение (16 бит)
* В команде задается смещение
  + Mov dx, [bx + si + $150]
* Номер сегмента находится в сегментном регистре
  + Какой сегментный регистр используется – зависит от того, какое действие выполняет процессор
* CS:IP задают полный адрес следующей команды
  + Те CS содержит номер сегмента, IP – смещение внутри сегмента
* Операнды в памяти по умолчанию задают смещение в сегменте DS
  + Если при адресации используется регистр BP, то сегмент SS
    - Mov dx, [bp – 4]
  + Можно переопределять используемый сегментный регистр явно:
    - Mov [es:bx + $953]
* Один и тот же байт может иметь несколько разных адресов
* При запуске программы настр. сегментные регистры на свободной области памяти, выбра. для запуска программы
* Например, COM-программа при запуске будет загружена в произвольный сегмент, но по смещению $0100, те по адресу

$XXXX:$0100

**Эффективный адрес**

**Эффективный адрес** – смещение операнда в памяти

* При обращении к операнду в памяти вычисляется выражение в квадратных скобках
  + Это значение и есть эффективный адрес

**mov** **bx**, 8

**mov** **si**, 6

**mov** **ax**, **[bx**+**si**+5**]**

**Команда LEA:**

* LEA:
  + Вычисляет эффективный адрес и помещает его в операнд-приемник
* Пример:
  + **lea** **dx**, **[bx** + **si** + 42**]** *; нет даже обращения в память*
* Операнды:
  + Приемник (первый) – всегда регистр
  + Источник – всегда память
* LEA:
  + Не выполняет обращение к памяти
  + Поэтому можно использовать для вычисления значения выражения:

**mov** **ebx**, 13

**mov** **esi**, 48

**lea** **eax**, **[ebx**\*4 + **esi** + 63**]** *; Для 32 бит*

* Не изменяет флагов!
* Смещение может быть задано:
  + Числом
  + Один из числовых регистров **(BX,BP,SI,DI)** + число
  + Регистр SX или BP + число
  + Регистр SI или DI + число