

# I like to move it!

Или команда MOV



# Команда MOV

- Команда MOV, хоть название ее и происходит от слова «move» (перемещать), на самом деле **не перемещает**, а **копирует** значение из источника в приемник:
- MOV приемник, источник



# Например...

```
mov ax,[number]
```

вносим значение переменной number  
в регистр AX

```
mov [number],bx
```

загрузить значение регистра BX  
в переменную number

```
mov bx,cx
```

занести в регистр BX значение  
регистра CX

```
.....
```

```
mov esi,edi
```

регистра EDI  
копировать значение регистра EDI  
в регистр ESI



# Внимание!

- Процессоры семейства x86 позволяют использовать в командах **ТОЛЬКО ОДИН КОСВЕННЫЙ аргумент**.
- Следующая команда копирования значения, находящегося по адресу `number_one`, в область памяти с адресом `number_two`, недопустима:
- `mov [number_two], [number_one]`

# Что же делать?

- Чтобы скопировать значение из одной области памяти в другую, нужно использовать промежуточный регистр:

```
mov ax, [number_one]    ;загружаем в AX 16-битное  
                        ;значение "number_one"  
mov [number_two], ax   ;а затем копируем его в переменную  
                        ;"number_two"
```

# Еще раз внимание!

- Оба операнда команды MOV должны быть одного размера:
- `mov ax, bl`
- ; НЕПРАВИЛЬНО! – Операнды разных ; размеров.

# Что же делать?

- Для копирования значения ВL в регистр АХ мы должны «расширить диапазон», то есть скопировать весь ВХ в АХ, а затем загрузить 0 в АХ:
- `mov ax, bx ; загружаем ВХ в АХ`
- `mov ah, 0 ;`
- "сбрасываем" верхнюю часть `; АХ`  
— записываем в нее 0



You like to **MOV** it?

