Far addresses

<u>Address of a memory location</u> = nr. of consecutive bytes from the beginning of the RAM memory and the beginning of that memory location.

Ex: the first element will have the address 0x00000000 (for flat memory model).

<u>Memory segment</u> = an interrupted sequence of memory locations, used for similar purposes during a program execution, a logical section of a program's memory, featured by its basic address, by its limit and by its type.

Ex: code segment contains machine instructions (1-15 bytes)

<u>Offset</u> = the address of a location relative to the beginning of a segment / the number of bytes between the beginning of that segment and that particular memory location

ex: data segment: a db 1 -> offset 0 (starting from $$)
b db 2 -> offset 1 (-"-).

<u>FAR address</u> = defines completely both the segment and the offset inside it

ex: mov EAX, [DS: a] -> moves into EAX a's FAR address

<u>segmentation</u> = memory management mechanism that decides the physical memory into variable-sized segments.

Ex: DS, CS, SS, ES.

<u>linear address</u> = (address computation) composed of base and offset (addresses) ~32 bits

Ex: $a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0 := b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 + o_7 o_6 o_5 o_4 o_3 o_2 o_1 o_0$

let base = 2000h , offset = 1000 h => linear address = 2000h + 1000h = 3000h flat memory model = the linear address has the base = 0

$a_7a_6a_5a_4a_3a_2a_1a_0 = 00000000 + o_7o_6o_5o_4o_3o_2o_1o_0$

ex: used by most of the modern operating systems : Windows

physical effective address = final result of segmentation plus paging eventually. The final address obtained by BIU points to physical memory (hardware) - at least 32 bits

direct addressing = addressing the memory when only a constant is present.

ex: mov eax, [a+4]

based addressing = if in the computing one of the base registers is present.

ex: mov eax,[edx]

scale - indexed addressing = if in the computing one of the index registers is present.

ex: mov ~~ax, [ebx+v+4]~~ eax, [2 · eax]

indirect addressing = a non direct addressing mode (based and/or indexed). at least one register specified between brackets.

ex: mov ax, [ebx +v+4].

near address = an address for which only the offset is specified, the segment address being implicitely taken from a segment register. A NEAR address is allways inside one of the 4 active segments.

ex: mov eax, [v].

register mode = if the required operand is a register.

mov eax, 17

The implicit rules for performing this association with an explicit specified offset operand are:

- CS for code labels target of the control transfer instructions. (jmp, call, ret, jz, etc).

  Ex: jmp there.

- SS in SIB addressing when using EBP or ESP as base (no matter of index or scale). Ex: mov AX, [ESP].

- DS for the rest of the data accesses