Friday Nov. 24th - 6pm - 8:30pm

16bits. offset specification formula:

$$\underbrace{[BX|BP]}_{\text{based addressing}} + \underbrace{[SI|DI]}_{\text{index addressing}} + \underbrace{[\text{const.}]}_{\text{direct addressing}}$$

```
mov ax, [eax + ebp * 4 - 7]

mov eax, [bx + si + 6]
```

```
a   db 17, -2, 0ffh, 'xyz'
    db 'a', -101, 251, -3, 7ah
    db .....
b   dw .....
lga dw $-a   => length in bites always
lga dw lga-a
mov eax, b-a
```

same as

Exam question:

The assembler / compiler has two main tasks:

1. To check the syntatic and correcness and validity of your source code
2. Generating the corresponding bytes for the instructions and directives met in the source code

```
mov eax, $-a
```

=> location counter in code section, can't use it here

```
mov eax, lga-a
```

```
lga equ lga-a
```

=> symbolic constant (not a variable => no memory allocation)

you can't write:

```
mov eax,[lga]
```

(syntax error because it's a constant)

but `mov eax, lga` works

```
Segment code
  start:
   jmp real_start:
     a db ...
     b dw ....
     c dd .....
  real_start:
```

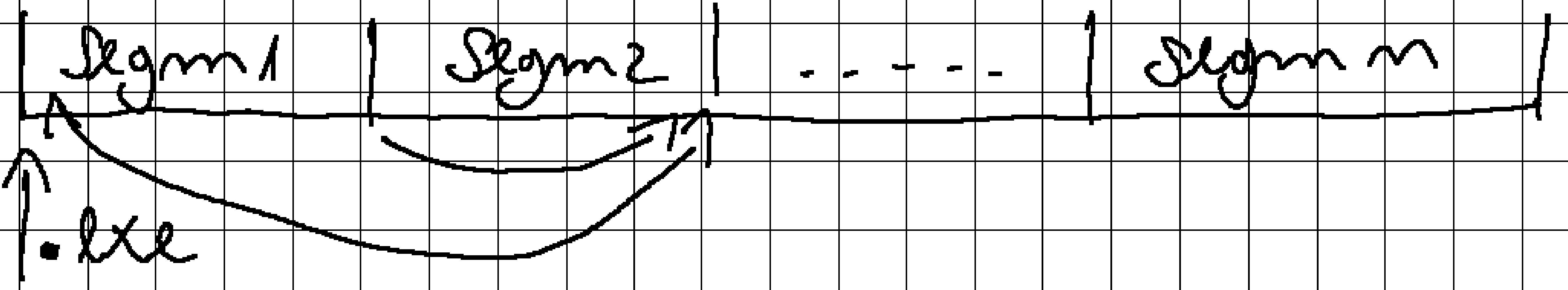
```
mov eax ,[ebx+2]

mov eax, ebx+2   => syntax error

mov eax ,[v+2]
mov eax ,[v]+2  => Syntax error
```

their values can't be determined at assembly time

At linking time:



mov eax, [v] ; mov eax, DWORD PTR
DS:[405000]

check lecture for explicit specified offset operand

mov eax, [ebx] ; — " — DS:[ebx]

mov eax, [ebp] ; — " — SS:[ebp]

mov eax, [ebp*2] ; mov eax, DWORD PTR
SS:[ebp + ebp]

mov eax, [ebp*3] ; — " — SS:[ebp+ebp*2]

mov eax, [ebp*4] ; — " — DS:[ebp*4]

mov eax, [ebx + esp] ; mov eax DWORD PTR [SS:esp+ebx]

mov eax, [esp + ebx] ; —— " ——

mov eax, [ebx + esp*2] ; - syntax error

mov eax, [ebx + ebp*2] ; - DS

- " - [ebx + ebp] ; — " — DS

- " - [ebp + ebx] ; — " — SS

base

takes them in order bcs of ambiguity
because both can be bases

[ebx*2 + ebp] - SS

[ebx*1 + ebp] - SS

[ebp*1 + ebx] - DS

→ base

[ebx*1 + ebp*1] - SS

first one with a scale is considered an index

[ebp*1 + ebx*1] - DS

jmp et1 ; jmp short 0001029