

## Curs 5

Segmente de date pe:

- 1) 16 biți:
  - CS (code segment)
  - DS (data segment)
  - SS (stack segment)
  - ES (extra segment)
- 2) 32 de biți (regiștrii segment adaptabili):
  - FS
  - GS
  - EIP (extended instruction pointer)

Denumirile provin de la programarea pe 16 biți.

Adresa unei locații – nr de octeți consecutivi dintre începutul memoriei RAM și începutul locației respective.

ES și DS sunt segmente pentru date.

**Orice program este împărțit în 4 segmente ES, CS, DS, SS**

Adresa e numărul de ordine a unei căsuțe din memorie.

Componenta ADR este cea care se ocupă de calculul adreselor.

Programatorii nu lucrează niciodată la nivelul codului sursă cu adrese fizice efective, ei lucrează cu așa-zisele specificații de adresă.

Noi ca și programatori lucrăm cu 2 numere:

ADR segment (16 biți) și offset (32 biți).

Segment logic:

- Început (adresa de bază)
- Sizeof-ul (dimensiunea)
- Sip (cod, date, stivă, extrasegment)

Întotdeauna combinația de regiștrii:

**CS:EIP** va reprezenta adresa instrucțiunii curente de reprezentat.

3 moduri de specificare a unui operand explicit:

- 1) Modul registru – mov eax, 17
- 2) Modul imediat – mov eax, 17
- 3) Modul de adresare la memorie, în acest caz offset-ul

Offset = [bază] + [index \* scală] + [constantă] ([ ] operatorul de dereferențiere.)

eax, ebx, ecx, edx, ebp, esi, edi sau esp baza

eax, ebx, ecx, edx, ebp, esi sau edi index

Scală = 1, 2, 4, 8.

Dacă avem numai [bază] și [index \* scală] se numește adresă indirectă (+ bazată + index scalată).

Dacă avem nu avem vreun registru între paranteze se numește adresă directă.

EBX = 17152

mov eax, ebx ; eax = 17152

mov eax, [ebx] 793456F0h

				F0h	56h	34h	79h
--	--	--	--	-----	-----	-----	-----

O variabilă = o celulă de memorie a cărei conținut este variabil, dar adresa constantă.