Arhitectura x86

De citit: Dandamudi
Capitolul 3: 3.1, 3.2

Modificat: 22-Oct-23

Cuprins curs 2-3

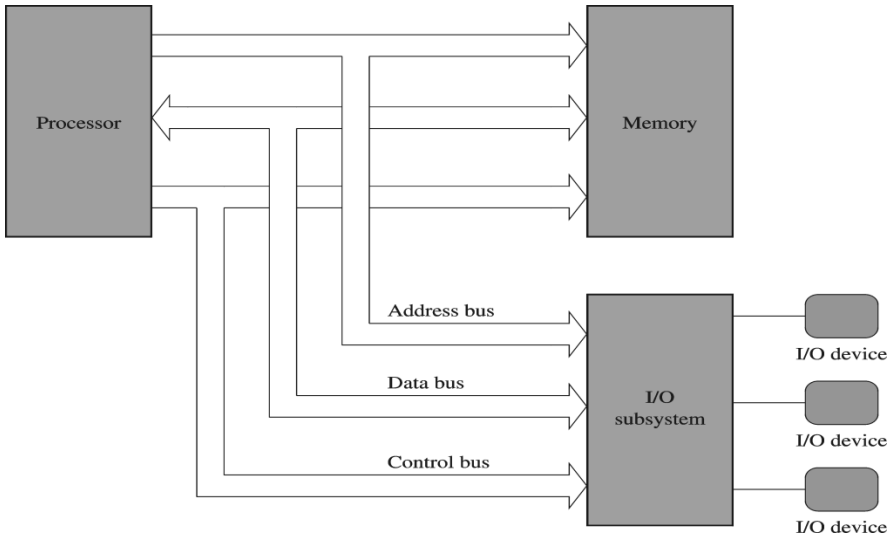
- Familia x86
- Registrele
  - Data
  - Pointer, index
  - Control
  - Segment
- Modul protejat
  - Registrele Segment
  - Descriptori de segment
  - Tabele de descriptori
  - Modele de segmentare

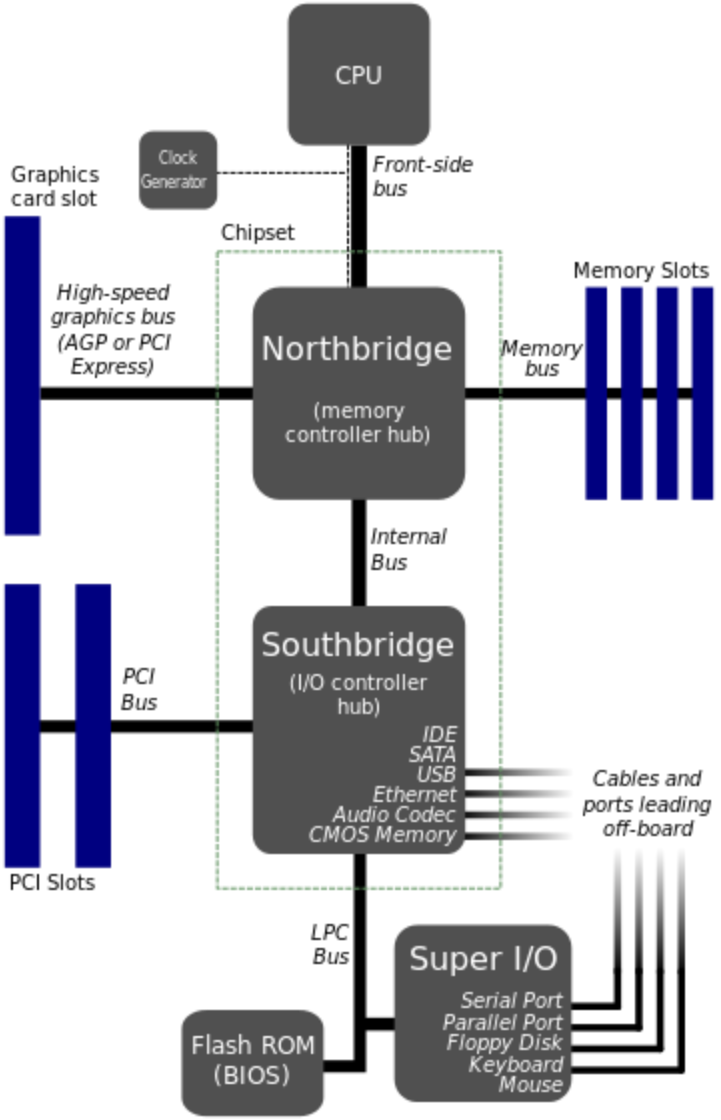
- Modul real
- Segmentare, Paginare
- Întreruperi
- Demo sasm, gdb
- Instrucțiuni mov, add, jmp

Istoricul procesoarelor Intel

Procesor	An	Frecvența	Tranzistoare	Registre	Bus date	Max addr
4004	1969	0.74	2.3K	4	12	4K
8080	1974	2	4.5K	8	16	64K
8086	1978	8	29K	16	16	1 MB
80386	1985	20	275K	32	32	4 GB
Pentium	1993	60	3.1M	32	40?	4GB
Pentium 4	2000	1500	42M	32		64GB
Core 2	2006	3000	291M	64		64GB
Core i7	2008	3400	1.4G	64		64GB
Xeon E5	2012	3600	5.5G	64		768GB

Arhitectura x86

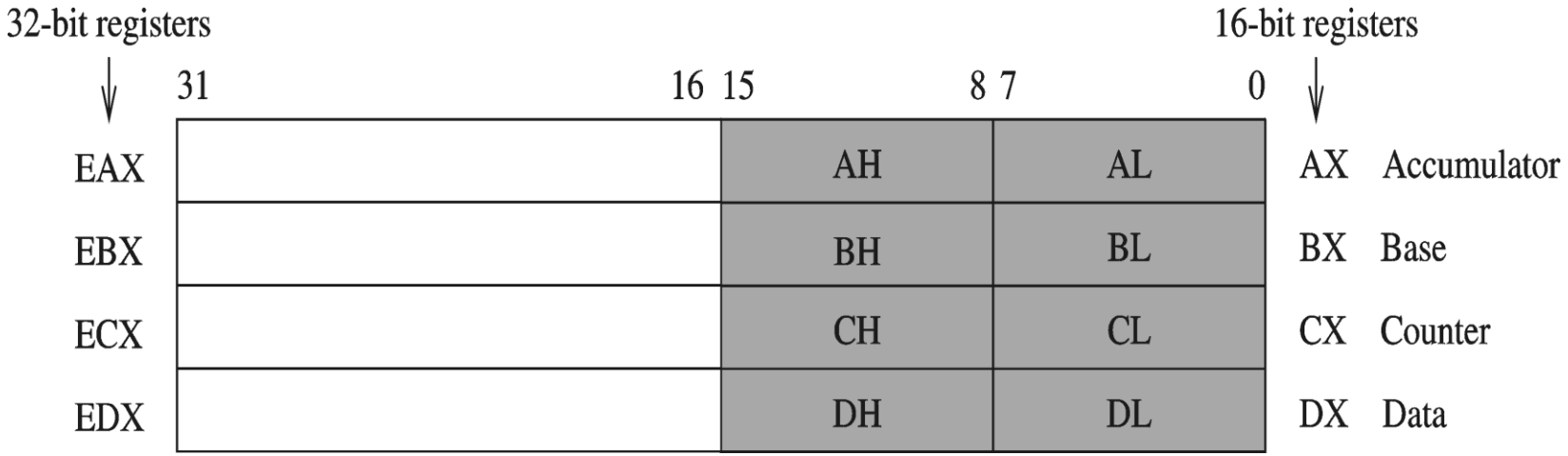




x86 registre pe 32 biți

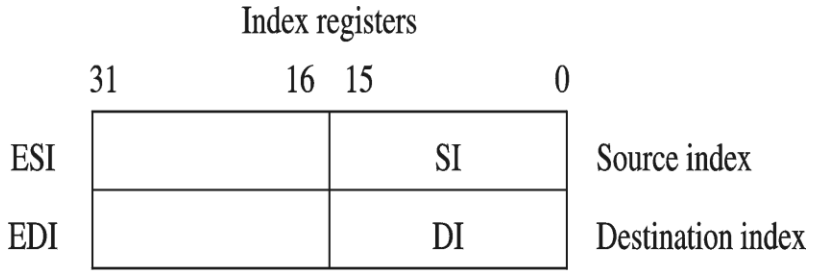
- Registre de 32 biți pot fifolosite
  - Pe 32 biți (EAX, EBX, ECX, EDX)
  - Pe 16 biți (AX, BX, CX, DX)
  - Pe 8 bit biți (AH, AL, BH, BL, CH, CL, DH, DL)

- Unele registre au utilizări speciale
  - ECX este numărator pentru instrucțiunea loop

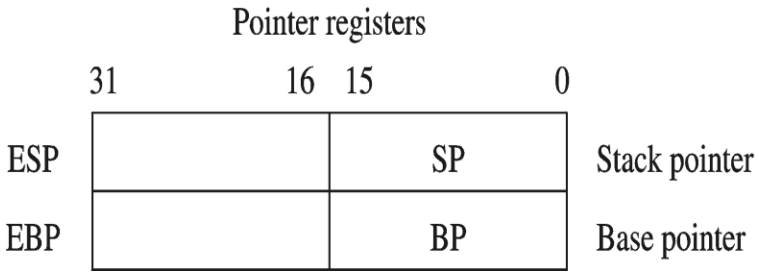


x86 registre pe 32 biți

- Două registre index
  - 16 sau 32 biți
  - Instrucțiuni pe stringuri
  - source (SI); destination (DI)
  - Pot fi folosite în scop general

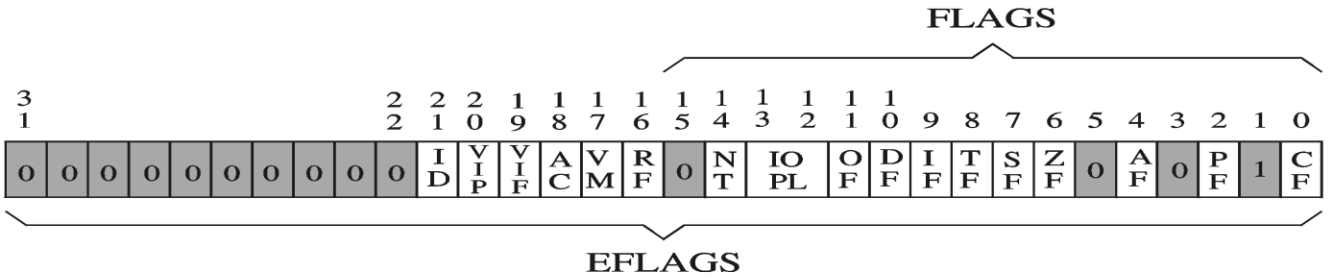


- Două registre pointer
  - 16 sau 32-biți
  - Exclusiv pentru stivă



x86 registre pe 32 biți

Flags register



Status flags

CF = Carry flag
PF = Parity flag
AF = Auxiliary carry flag
ZF = Zero flag
SF = Sign flag
OF = Overflow flag

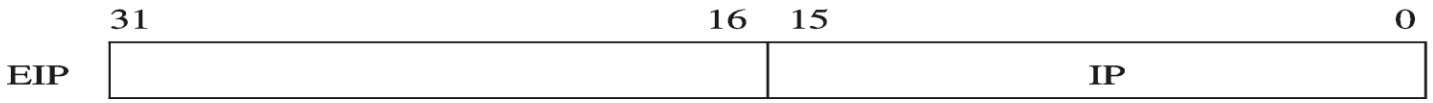
Control flags

DF = Direction flag

System flags

TF = Trap flag
IF = Interrupt flag
IOPL = I/O privilege level
NT = Nested task
RF = Resume flag
VM = Virtual 8086 mode
AC = Alignment check
VIF = Virtual interrupt flag
VIP = Virtual interrupt pending
ID = ID flag

Instruction pointer



EFLAGS (Indicatorii de stare)

- CF (Carry Flag) - indicator de transport - reflecta transportul in exterior al bitului cel mai semnificativ al rezultatului operatiilor aritmetice. Astfel, acest indicator poate fi folosit in cazul operatiilor in dubla precizie. Valoarea CF = 1 semnifica fie transport la adunare fie imprumut la scadere. De asemenea, indicatorul CF este modificat si de instructiunile de deplasare si rotatie.
- PF (Parity Flag) - indicator de paritate - este 1 daca rezultatul are paritate para (contine un numar par de biti 1). Acest indicator este folosit de instructiunile de aritmetica zecimala.
- AF (Auxiliary Carry Flag) - indicator de transport auxiliar - este 1 daca a fost transport de la jumatatea de octet inferioara la jumatatea de octate superioara (de la bitul 3 la bitul 4). Acest indicator este folosit de instructiunile de aritmetica zecimala.
- ZF (Zero Flag) - indicatorul de zero - este 1 daca rezultatul operatiei a fost zero.
- SF (Sign Flag) - indicatorul de semn - este 1 daca cel mai semnificativ bit al rezultatului (MSb) este 1, adica in reprezentarea numerelor in complement fata de 2 (C2) rezultatul este negativ (are semn -).
- OF (Overflow Flag) - indicatorul de depasire aritmetica (a gamei de valori posibil de reprezentat) - este 1 daca dimensiunea rezultatului depaseste capacitatea locatiei de destinatie si a fost pierdut un bit (indica la valorile cu semn faptul ca se "altereaza" semnul).

EFLAGS(Indicatorii de control)

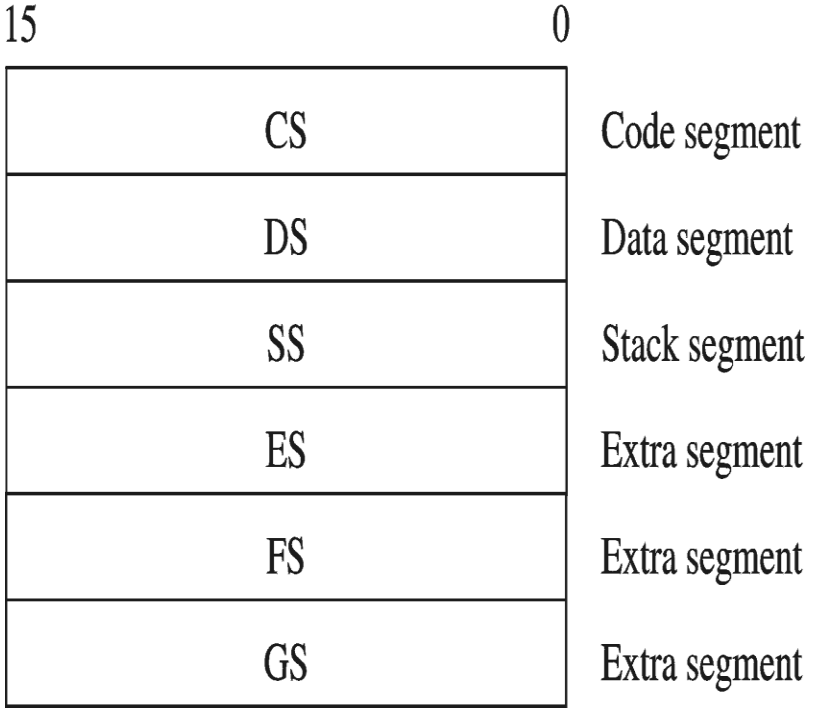
- DF (Direction Flag) – este utilizat de instrucțiunile pe şiruri şi specifică direcția de parcurgere a acestora:
  - 0 – şirurile se parcurg de la adrese mici spre adrese mari;
  - 1 – şirurile sunt parcurse invers.
- IF (Interrupt Flag) – acest indicator controlează acceptarea semnalelor de întrerupere externă. Dacă IF = 1 este activat sistemul de întreruperi, adică sunt acceptate semnale de întrerupere externă (mascabile, pe linia INTR); altfel, acestea sunt ignorate. Indicatorul nu are influență asupra semnalului de întrerupere nemascabilă – NMI.
- TF (Trace Flag) – este utilizat pentru controlul execuției instrucțiunilor în regim pas cu pas (instrucțiune cu instrucțiune), în scopul depanării programelor. Dacă indicatorul este 1, după execuția fiecărei instrucțiuni se va genera un semnal de întrerupere intern (pe nivelul 1). Evident, execuția secvenței de tratare a acestei întreruperi se va face cu indicatorul TF = 0.

x86 registre pe 32 biți

- registre de control
  - EIP
    - Instruction pointer (instrucțiunea curentă)
  - EFLAGS
    - Status flags
      - Se actualizează după operații aritmetice/logice
    - Direction flag
      - Forward/backward direcția copierii
    - System flags
      - IF : activare intreruperi
      - TF : Trap flag (pentru debugging)

x86 registre pe 32 biți

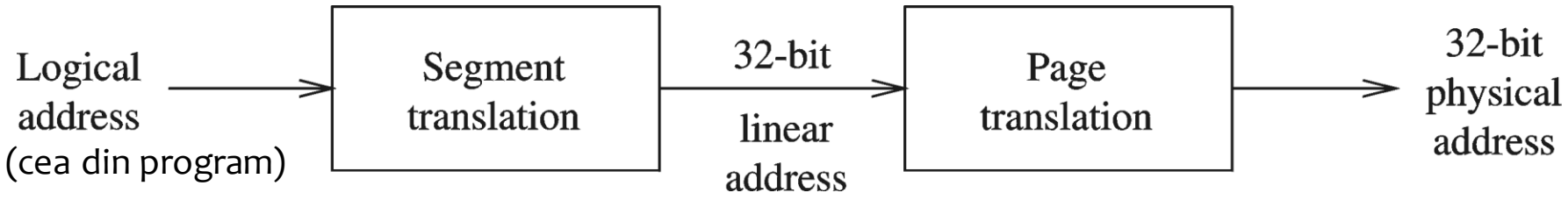
- Registre segment
  - 16 biți
  - Memoria segmentată
  - Conținut distinct
    - » Code
    - » Data
    - » Stack



Nu le folosim acest semestru :-)

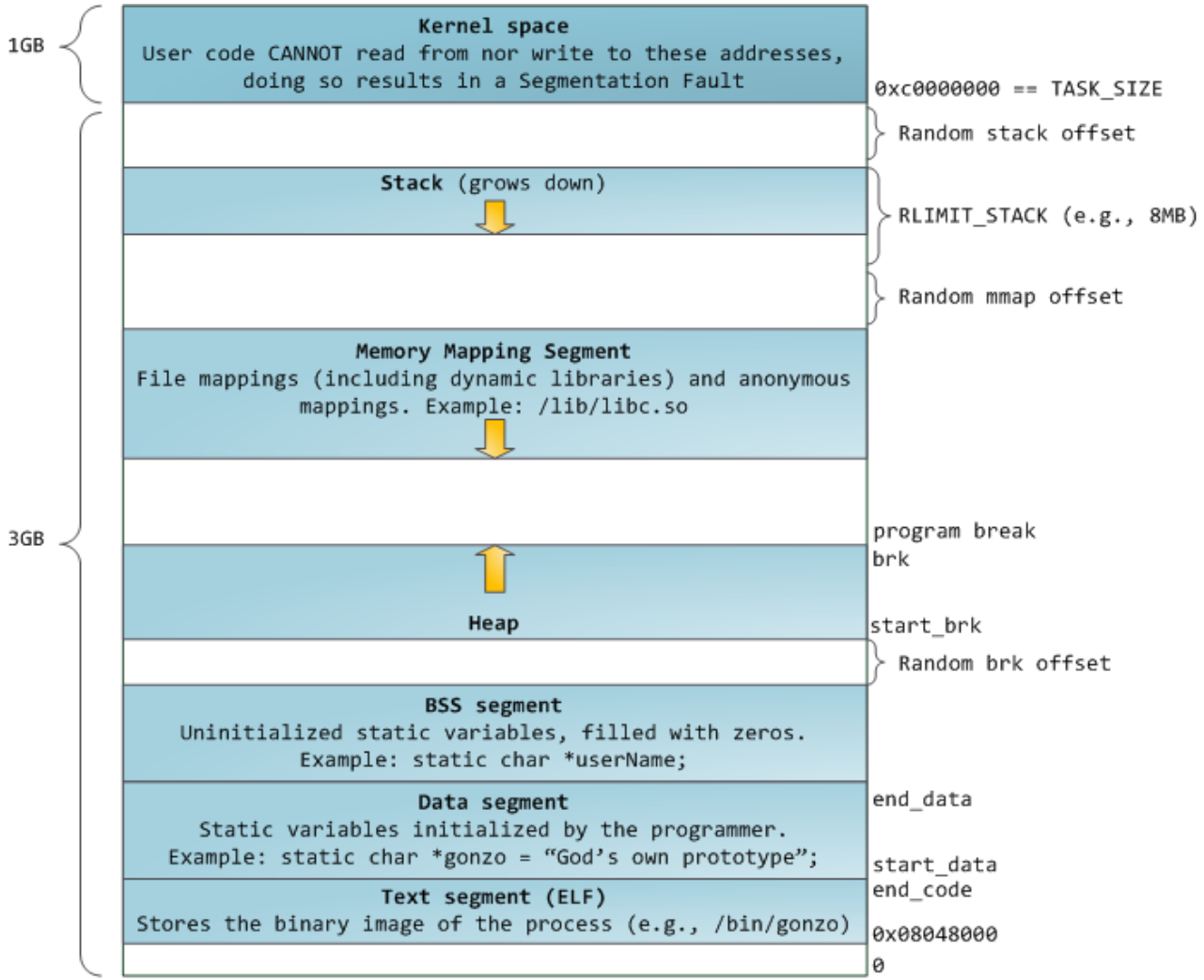
Adrese logice și fizice

- Segmentarea&paginarea = translatare adrese 32 biți
- Segmentarea: adrese logice → adrese lineare
- Paginarea: adrese lineare → adrese fizice
- Segmentare, paginare → cursurile SO, SO2



IOCLA, PC, SD **SO, SO2** **CN1, CN2**

Imaginea unui proces în memorie

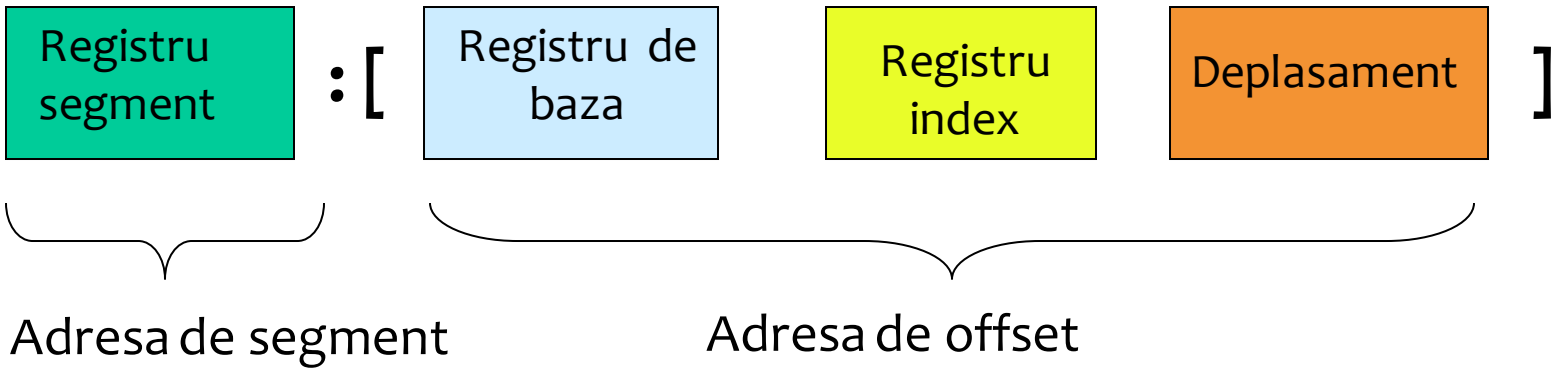


Adrese logice (în program)

http://duartes.org/gustavo/blog/post/anatomy-of-a-program-in-memory/

Adresarea memoriei

calculul adresei logice



Adresarea memoriei

$$\left\{\begin{matrix} CS: \\ DS: \\ SS: \\ ES: \\ FS: \\ GS: \end{matrix}\right\} \left[\left\{\begin{matrix} EAX \\ EBX \\ ECX \\ EDX \\ ESP \\ EBP \\ ESI \\ EDI \end{matrix}\right\} \right] + \left[\left\{\begin{matrix} EAX \\ EBX \\ ECX \\ EDX \\ EBP \\ ESI \\ EDI \end{matrix}\right\} * \left\{\begin{matrix} 1 \\ 2 \\ 4 \\ 8 \end{matrix}\right\} \right] + [\text{displacement}]$$

```
mov eax, [mybuffer + ebx + esi*4 + 9]
; ebx = bază
; esi = index
; mybuffer + 9 = deplasament
```

Demo

- Programul bc – conversii numerice (obase, ibase)
- https://github.com/systems-cs-pub-ro/iocla
- `curs-03-x86/demo/` – gdb hello
  - Registrele EAX, AX, AH, AL
  - EFLAGS
  - EIP
  - Instrucțiunile MOV, ADD, JMP
  - Comenzile b main, r, n
  - set $eax = 0xffffffff
  - set $eip = main
  - Decomentați instrucțiunea jmp și reasamblați
- Atenție la ~/.gdbinit

Intrebări?