

Observatii:

* je <=> cu jz (pentru ca cmp pune in tmp = op1 - op2 caz in care daca sunt egale da 0 adica ZF= 1 adica jz da true)
* a(b ,c ,d)

a – adresa de memore/constanta

b – registru

c – registru (%ecx)

d – constanta

v[2] = 5

caz a = 0

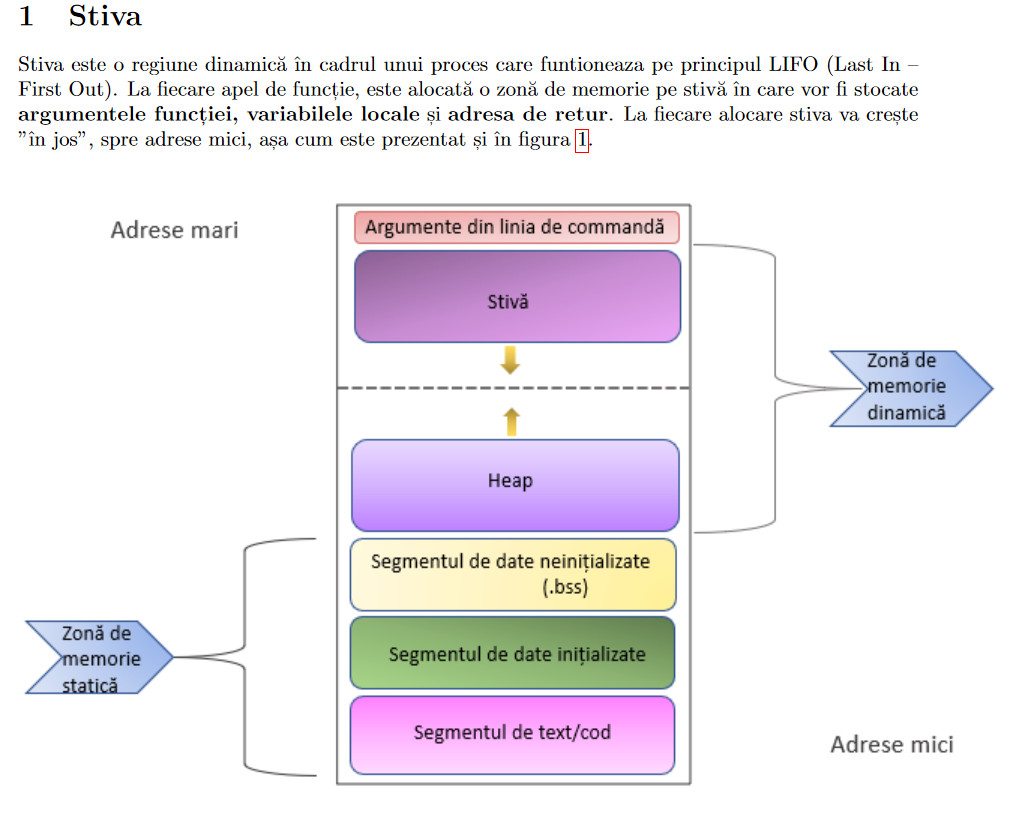
mov $v, %edi

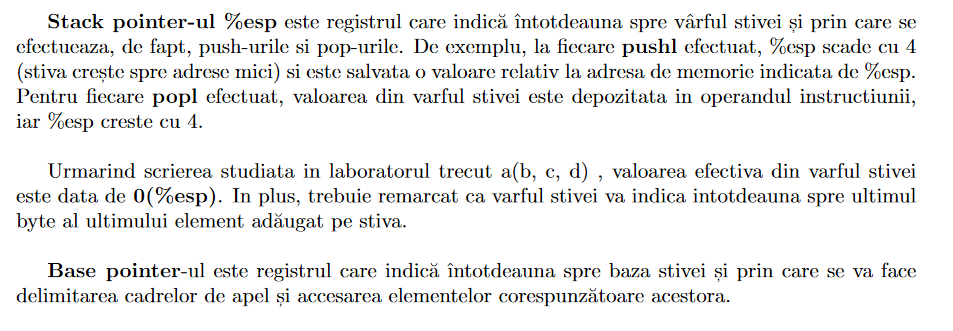
mov $2, %ecx

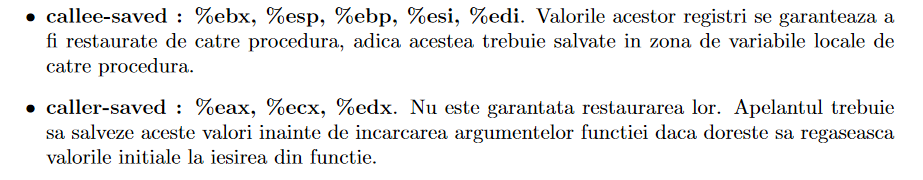
movl $5, (%edi, %ecx, 4)

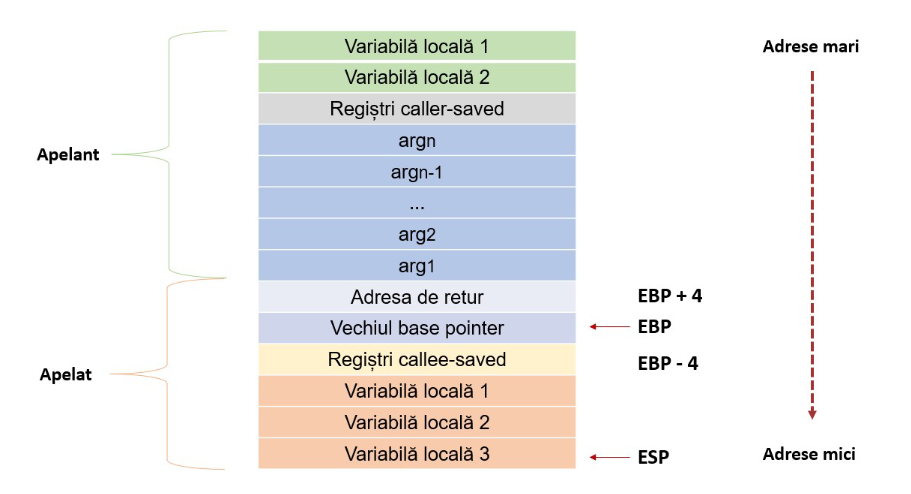
caz b = 0

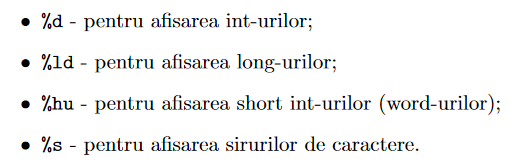
movl $5, v( ,%ecx,4)











Observatii:

* pot fi adaugate pe stiva doar word-uri si long-uri
* pusha (pune toate valorile registrilor pe stiva, si la descarcare pastreaza valorile)
* Cand stiva creste esp ul scade, ebp ramane la aceeasi valoare
* 0(%esp) va fi mereu adresa urmatoarei instructiuni, deci vom pleca de la 4(%esp)
* In cazul in care se doreste returnare, valorile vor fi depozitate in aceasta ordine in functie de numarul lor in %eax, %ecx, %edx si ulterior prin varful stivei