Daca executam urmatorul main vom obtine segmentation fault; care este semnificatia acestei erori? main:

movl $100, %esp

pushl %eax

movl $0, %eax

xorl %ebx, %ebx

int $0x80

**Solutie:** Registrul %esp primeste o adresa de memorie dintr-o zona de adrese mici, in care programul nostru nu are drepturi. Cand se incearca pushl %eax, se incearca o completare de valoare la adresa nepermisa de memorie, si este aruncat segmentation fault.

**Intrebare bucla infinita: de tipul et\_loop:**

Cmp $0, %ecx

Jle et\_exit

Incl %ecx

Jmp et\_loop

se va executa for-ul pentru 0xae2a, 0xae2b, 0xae2c, ..., pana la 0x7fffffff = 0b01...1 = 2147483647 (cel mai mare intreg cu semn reprezentabil pe 32 de biti), iar apoi va ajunge la -2147483648 (0b10...0), unde nu se va mai respecta conditia de a ramane in cadrul structurii repetitive.

Teorie:

Eax(32)=ax(16),ah(8),al(8)

Analog pt ebx, ecx si edx

EAX,ECX,EDX TREBUIE RESTAURATI DUPA APEL

Tipuri de date:

1. Byte 1 byte, 8 biti
2. Single 4 bytes, 32 de biti
3. Word 2 bytes 16 biti
4. Long 4 bytes, 32 de biti
5. Quad 8 bytes, 64 de biti
6. Ascii -siruri de caractere fara terminator de sir
7. Asciz siruri de caractere cu terminator de sir
8. Space spatiu de memorie cu dimensiune variabila(.space 4 spatiu de 4 bytes=32 biti)

Salturi

jnc jump daca nu este carry setat

jno jump daca nu este overlow setat

jnz jump daca nu este zero setat

jns jump daca nu este sign setat

pt nr fara semn:

jb jump if below (op1 < op2)

jbe jump if below or equal (op1 <= op2)

ja jump if above (op1 > op2)

jae jump if above or equal (op1 >= op2)

pt numere cu semn:

jl jump if less than (op1 < op2)

jle jump if less than or equal (op1 <= op2)

jg jump if greater than (op1 > op2)

jge jump if greater than or equal (op1 >= op2)

jne jump if not equal (op1 ! = op2)

ret Aceasta instructiune realizeaza un salt la adresa din varful stivei. Cum pe parcursul procedurii apelate toate elementele adaugate pe stiva sunt si eliminate, la final in varful stivei se va afla chiar adresa de retur retinuta prin instructiunea call.

Creare cadru de apel procedura:

pushl %ebp

movl %esp, %ebp

#codul din corpul procedurii

popl %ebp

ret

%d - pentru afisarea int-urilor;

• %ld - pentru afisarea long-urilor;

• %hu - pentru afisarea short int-urilor (word-urilor);

• %s - pentru afisarea sirurilor de caractere.