

### **Cod apel (APELATORUL):**

- a). Salvare resurse volatile (EAX, ECX, EDX, EFLAGS)
- b). Transmitere parametri
- c). Efectuare apel cu salvare adresa de revenire

### **Cod de intrare (APELATUL):**

a). Creare stackframe nou      PUSH EBP,  
   MOV EBP, ESP

- b). Alocare spațiu variabile locale      SUB ESP,nr\_octeti
- c). Salvare resurse nevolatile posibil a fi modificate

### **Cod de ieșire (APELATUL):**

- a). Restaurare regiștri nevolatili
- b). Eliberare spațiu variabile locale      [ADD ESP, nr\_octeti\_var\_locale] – menționată aici doar ca revers al pct b) din codul de intrare , dar neobligatorie deoarece eliberarea cadrului de stivă (mov esp, ebp) include oricum dpdv practic si această etapă.
- c). Eliberare cadru de stivă      MOV ESP, EBP (daca stim exact dimensiunea cadrului de stiva, ADD ESP, sizeof(stackframe) rezolva in mod similar...)  
   POP EBP  
   (a, b c – reversul codului de intrare)
- d). Revenirea din subprogram (RET) și scoaterea de pe stivă a parametrilor (daca este de tip STDCALL)      -      (reversul b + c din codul de apel)

A mai rămas de efectuat reversul pct a) din codul de apel. Este sarcina APELATORULUI să o facă alături de eventuala scoatere de pe stivă a parametrilor (daca este de tip CDECL). Aceste actiuni trebuie sa se efectueze de catre codul apelant imediat dupa incheierea apelului.