Programare multi-modul în limbajul de asamblare

* Avem mai multe fisiere cu extensia .asm care trebuie asamblate separat  
  nasm –fobj modul1.asm  
  …  
  nasm –fobj modulN.asm
* Se va crea un singur fisier executabil cu numele modul1.exe prin link-editarea tuturor modulelor obj obtinute in urma asamblarii:  
  alink –oPE –subsys console –entry start modul1.obj modul2.obj … modulN.obj

Important:

* Un singur modul va contine programul principal (eticheta start)
* Restul modulelor descriu proceduri care vor fi apelate din programul principal
* Folosind **global** se poate exporta o procedura / o variabila din modul curent pentru a putea fi folosit in celelalte module
* Pentru a importa in celelalte module o procedura / o variabila se foloseste cuvantul cheie **extern**
* **Constantele nu se pot exporta!**

Cum se transmit parametrii unei proceduri?

1. Prin intermediul registrilor   
   Dezavantaj: exista un numar limitat de registrii care pot fi folositi pentru transmiterea parametrilor
2. Prin intermediul variabilelor globale  
   Dezavantaj: nu se respecta principiul modularitatii
3. Prin intermediul stivei  
   Avantaje: solutie flexibila si preferata

Sa se rezolve x=a+b.

1. Se transmit parametrii pentru procedura adunare prin intermediul registrilor

|  |  |
| --- | --- |
| MAIN.asm | MODUL.asm |
| bits 32  global start  extern exit import exit msvcrt.dll **extern adunare**  segment data use32 class=data **public**  a db 1 b db 2 x db 0  segment code use32 class=code **public** start:  mov bl,[a] mov bh,[b] call adunare  mov [x],al   push dword 0 call [exit] | bits 32 **global adunare**  segment data use32 class=data **public**  segment code use32 class=code **public**  adunare:  mov al,bl add al,bh  ret |

1. Se transmit parametrii prin intermediul variabilelor globale

|  |  |
| --- | --- |
| MAIN.asm | MODUL.asm |
| bits 32  global start  extern exit import exit msvcrt.dll extern adunare  **global a** **global b** **global x**  segment data use32 class=data public a db 1 b db 2 x db 0  segment code use32 class=code public start:  call adunare  push dword 0 call [exit] | bits 32 global adunare **extern a,b,x**  segment code use32 class=code public  adunare:  mov al,[a] add al,[b] mov [x],al  ret |

1. Se transmit parametrii prin intermediul stivei

|  |  |
| --- | --- |
| MAIN.asm | MODUL.asm |
| bits 32  global start  extern exit import exit msvcrt.dll  extern adunare  segment data use32 class=data public a db 1 b db 2 x db 0  segment code use32 class=code public start:  **mov eax,0** **mov al,[a]** **push eax**  **mov eax,0** **mov al,[b]** **push eax**  **mov eax,0** **mov al,[x]** **push eax**  call adunare  **pop eax;** se ia de pe stiva valoarea lui x calculata in urma adunarii **mov [x],al** **add esp,4\*2**  push dword 0 call [exit] | bits 32  global adunare  segment code use32 class=code public  adunare:  **; STIVA** **; [ESP] <- adresa de revenire in main** **; [ESP+4] <- valoarea lui x** **; [ESP+8] <- valoarea lui b** **; [ESP+12] <- valoarea lui a**  **mov eax, [ESP+12]** **add eax, [ESP+8]** **mov [ESP+4], eax**  ret |

Probleme:

1. Scrieti un program care concateneaz 2 siruri prin apelul unei functii scrise intr-un modul secundar si apoi afiseaza rezultatul pe ecran:

|  |  |
| --- | --- |
| MAIN.asm | MODUL.asm |
| Bits 32  Global start  Extern exit  Import exit msvcrt.dll  Extern printf  Import printf msvcrt.dll  Extern concatenare  Segment data use32 class=data public  Sir1 db “abcd”, 0  Sir2 db “efg”, 0  Rez times ($-sir1-1) db 0  Segment code use32 class=code public  Start:  Push dword sir1  Push dword sir2  Push dword rez  Call concatenare  Add esp, 4 \* 3  Push dword rez  Call [printf]  Add esp, 4 \* 1  Push dword 0  Call [exit] | Bits 32    Global Concatenare        Segment code use32 class=code public    Concatenare:    ;[esp] - adresa de revenire    ;[esp + 4] - adresa lui rez    ;[esp + 8] - adresa lui sir2    ;[esp + 12] - adresa lui sir1        Mov esi, [esp + 12]    Mov edi, [esp + 4]        Et\_loop:  lodsb  stosb  cmp al,0    Jne Et\_loop        Dec edi    Mov esi, [esp + 8]        Et\_loop2:    lodsb  stosb    Cmp al, 0    Jnz Et\_loop2        ret |

1. Se da un numar a reprezentat pe 32 de biti fara semn. Sa se afiseze suma cifrelor acestui numar.

|  |  |
| --- | --- |
| MAIN.asm | MODUL.asm |
| Bits 32  Global start  Extern exit  Import exit msvcrt.dll  Extern printf  Import printf msvcrt.dll  Extern suma\_cifre  Segment data use32 class=data public  X dd 1234  Suma resd 1  format db "suma = %d", 0  Segment code use32 class=code public  start:  Push dword [X]  Push dword Suma  Call suma\_cifre  Add esp, 4 \* 2  Push dword [Suma]  Push dword format  Call [printf]  Add esp, 4 \* 2  Push dword 0  Call [exit] | Bits 32  Global suma\_cifre  Segment data use32 class=data public  Zece dd 10  Segment code use32 class=code public  suma\_cifre:  ;[ESP] - adresa de revenire  ;[ESP+4] - adresa lui "Suma"  ;[ESP+8] - valoarea lui "X"  Mov EBX, 0  Mov eax, [ESP+8]  Repeta:  Mov edx, 0  Div dword [zece]  Add ebx, edx  Cmp eax, 0  Jnz repeta  Mov esi, [esp+4]  Mov [esi], ebx  ret |
|  |  |

1. Se citeste un cuvant de la tastatura. Criptati cuvantul citit adunand 20 la fiecare caracter. Afisati rezultatul obtinut pe ecran.

|  |  |
| --- | --- |
| MAIN.asm | MODUL.asm |
| Bits 32  Global start  Extern exit  Import exit msvcrt.dll  Extern printf  Extern scanf  Import printf msvcrt.dll  Import scanf msvcrt.dll  Extern criptare  Segment data use32 class=data public  Cuvant times 100 db 0  Format db “%s”, 0  Cuvant\_criptat times 100 db 0  Segment code use32 class=code public  Start:  ;scanf(format,cuvant)  Push dword cuvant  Push dword format  Call [scanf]  Add esp, 4\*2  Push dword cuvant  Push dword cuvant\_criptat  Call criptare  Add esp, 4\*2  Push dword cuvant\_criptat  Call [printf]  Add esp 4\*1  Push dword 0  Call [exit] | Bits 32  Global criptare  Segment data use32 class=data public  Segment code use32 class=code public  Criptare:  ; [ESP] adresa de revenire  ; [ESP+4] adresa rezultatului  ; [ESP+8] adresa cuvantului  Mov EAX, [ESP+8]  Mov esi, [ESP+4]  Repeta:  Mov BL, [EAX]  Cmp BL, 0  Je out  ADD BL, 20  Mov [esi], BL  INC EAX  INC ESI  Jmp Repeta  Out:  ret |

Se dau a,b,c,d in segmentul de date, dublucuvinte.

Doresc sa apelez in modulul principal functia f(d,a,c,b) si sa transmit

catre modul care are descrisa functia f parametrii:

- a ca variabila globala  
- b prin registru

- c si d prin stiva.

Cum arata secventa de instructiuni din programul principal care apeleaza functia f (inclusiv import-urile si export-urile de simboluri)?

Cum arata stiva la intrarea in functia f? Care este codul functiei f, stiind ca f calculeaza expresia a-b+c-d

Tema: Se dau doua siruri de caractere. Sa se afiseze sirul cu numar maxim de caractere speciale (orice caracter in afara de litere sau cifre).