**Arhitectura Sistemelor de Calcul**

**Seminar 5**

Apeluri de functii din biblioteci

Call [nume\_functie]

1. Afisarea pe ecran

printf(format, val1, val2,...)

* sir de caractere
* specificatori: %d, %u, %x, %s, ...

1. Citirea de la tastatura

scanf(format, var1, var2, ...)

* sir de specificatori

1. Sa se scrie un program care va afisa pe ecran mesajul “n=” si care citeste un nr de la tastatura

bits 32

global start

extern exit, printf, scanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

segment data use32 class=data

n dd 0

format db "%d", 0

mesaj db "n=", 0

segment code use32 class=code

start:

push dword mesaj

call [printf]

add esp, 4

push dword n

push dword format

call [scanf]

add esp, 4\*2

push dword 0

call [exit]

2. Sa se scrie un program in limbaj de asamblare care citeste 2 numere a si b, calculeaza suma lor si afiseaza rezultatul pe ecran

bits 32

global start

extern exit, printf, scanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

segment data use32 class=data

a dd 0

b dd 0

format db "%d %d", 0

format\_suma db "suma = %d", 10, 0 ; “<<>>”, 10, 0 = \n la final

mesaj db "gata", 0

segment code use32 class=code

start:

; scanf( "%d, %d", a, b)

push dword b

push dword a

push dword format

call [scanf]

add esp, 4\*3

;eax = a + b

mov eax, [a]

add eax, [b]

;printf("suma = %d\n", eax)

push eax

push dword format\_suma

call[printf]

add esp, 4\*2

;printf("gata")

push dword mesaj

call [printf]

add esp, 4

push dword 0

call [exit]

**3. Operatii cu fisiere text**

3.1. Deschiderea unui fisier

descriptor = fopen (nume\_fisier, mod\_acces)

mod\_acces = “r”, “w”, “a”, “r+”, “a+”, “w+”

3.2 Operatii de citire/scriere

fprintf (descriptor, format, val1, ...)

fscanf (descriptor, format, var1, ...)

fwrite (string, size, count, descriptor)

fread(string, size, count, descriptor) = nr caracterelor efectiv citite

3.3 Inchiderea unui fisier

fclose(descriptor)

;remove(nume)

;rename(old, new)

;perror(mesaj) ; eroare:...

3. Se citeste continutul unui fisier (a.txt). Se adauga 1 la fiecare octet citit si apoi se scriu octetii rezultati intr-un fisier nou (b.txt). Se redenumeste la finalul scrierii fisierul (b.txt) in (a.txt) si se sterge fisierul (b.txt) din folderul curent.

bits 32

global start

extern exit, fprintf, fscanf, remove, rename, perror, fopen, printf, fread, fclose

import exit msvcrt.dll

import fprintf msvcrt.dll

import fscanf msvcrt.dll

import remove msvcrt.dll

import rename msvcrt.dll

import perror msvcrt.dll

import fopen msvcrt.dll

import printf msvcrt.dll

import fread msvcrt.dll

import fclose msvcrt.dll

segment data use32 class=data

numefis1 db "a.txt", 0

numefis2 db "b.txt", 0

mod1 db "r", 0

mod2 db "a", 0

desc1 dd 0

desc2 dd 0

buffer db 1

format db "%c", 0

mesaj db "eroare", 0

segment code use32 class=code

start:

;fopen(numefis1, mod1)

push mod1

push numefis1

call[fopen]

add esp, 4\*2

cmp eax, 0

je afisare\_eroare

mov [desc1], eax

;fopen(numefis2, mod2)

push mod2

push numefis2

call [fopen]

add esp, 4\*2

cmp eax, 0

je afisare\_eroare

mov [desc2], eax

;fread (buffer, 1, 1, desc1)

repeta:

push dword [desc1]

push dword 1

push dword 1

push dword buffer

call[fread]

add esp, 4\*4

cmp eax, 0

je gatacitire

inc byte [buffer]

mov eax, 0

mov al, [buffer]

;printf(desc2, format, eax)

push eax

push format

push desc2

call[fprintf]

add esp, 4\*3

jmp repeta

gatacitire:

;fclose(desc1)

push dword [desc1]

call [fclose]

add esp, 4

;fclose(desc2)

push dword [desc2]

call[fclose]

add esp, 4

;remove (numefis1)

push dword numefis1

call [remove]

add esp, 4

;rename (numefis2, numefis1)

push dword numefis1

push dword numefis2

call[rename]

add esp, 4\*2

final:

push dword 0

call [exit]

afisare\_eroare:

push dword mesaj

call[fprintf]

add esp, 4

### nu merge

4. Sa se deschida fisierul (in.txt) care contine exact 16 octeti si sa se afiseze pe ecran acei octeti din fisier care se afla pe pozitiile corespunzatoare bitilor 1, din reprezentarea binara a numarului n citit

n=F2A1h = 1111 0010 1010 0001b

in.txt = Azi\_avem\_seminar

* Avmsinar

bits 32

global start

extern exit, fprintf, fscanf, remove, rename, perror, fopen, printf, fread, fclose, scanf

import exit msvcrt.dll

import fprintf msvcrt.dll

import fscanf msvcrt.dll

import remove msvcrt.dll

import rename msvcrt.dll

import perror msvcrt.dll

import fopen msvcrt.dll

import printf msvcrt.dll

import fread msvcrt.dll

import fclose msvcrt.dll

import scanf msvcrt.dll

segment data use32 class=data

n dd 0

format\_n db "%x", 0

fis1 db "in.txt", 0

mod\_acces db "r", 0

desc dd 0

buffer db 0

format\_afisare db "%c", 0

segment code use32 class=code

start:

;scanf( "%x", n)

push dword n

push dword format\_n

call [scanf]

add esp, 2\*4

;fopen (fis1, mod\_acces)

push dword mod\_acces

push dword fis1

call[fopen]

add esp, 2\*4

cmp eax, 0

je final

mov [desc], eax

mov ecx, 16

repeta:

;fread (buf, 1, 1, desc)

push dword [desc]

times 2 push dword 1

push dword buffer

call [fread]

add esp, 4\*4

shr dword [n], 1

jnc next

;printf(format\_afisare, eax)

mov eax, 0

mov al, [buffer]

push eax

push format\_afisare

call[printf]

add esp, 4\*2

next:

loop repeta

final:

push dword 0

call [exit]