Seminar 7

1. Se citesc de la tastatura nume de fisiere pana cand se introduce cuvantul “end”. Sa se gaseasca fisierul cu numar maxim de litere mari si sa se afiseze pe ecran numele si pe linie noua continutul fisierului.

|  |  |
| --- | --- |
| data segment | code segment |
| format\_citire “%s”, 0  nume\_fisier times 255 db 0  end db “end”, 0  max\_nume\_fisier times 255 db 0  max\_litere\_mari dd 0 | start:  repeta:  push dword nume\_fisier  push dword format  call [scanf]  add esp, 2\*4  mov esi, nume\_fisier  mov edi, end  mov ecx, 4  cld  comp:  cmpsb  jne final\_comp  loop comp  jmp afiseaza\_max  final\_comp:  push dword nume\_fisier  call numara\_litere\_mari  add esp, 1\*4  cmp eax, [max\_litere\_mari]  jna repeta  mov [max\_litere\_mari], eax  mov ecx, 255  mov esi, nume\_fisier  mov edi, max\_nume\_fisier  cld  rep movsb  jmp repeta  afiseaza\_max:  push dword max\_nume\_fisier  call afiseaza\_fisier  add esp, 1\*4 |

modul1.asm

|  |
| --- |
| stiva |
| adresa de revenire – esp |
| nume\_fisier – esp + 4 |

D.S.

mod\_acces db “r”, 0

descriptor dd -1

buffer db 0

nr\_litere dd 0

C.S.

numar\_litere\_mari:

mov dword [nr\_litere], 0

mov ebx, [esp+4]

push dword mod\_acces

push ebx

call [fopen]

add esp, 2\*4

cmp eax, 0

je final

mov [descriptor], eax

repeta:

push dword [descriptor]

push dword 1

push dword 1

push dword buffer

call [fread]

add esp, 4\*4

cmp eax, 0

je inchide\_fisier

cmp byte [buffer], ‘A’

jb repeta

cmp byte [buffer], ‘Z’

ja repeta

inc dword [nr\_litere]

jmp repeta

inchide\_fisier:

push dword [descriptor]

call [fclose]

add esp, 4\*1

final:

mov eax, [nr\_litere]

modul2.asm

D.S.

format\_nume db “%s”, 10, 13, 0

format\_c db “%c”, 0

mod\_acces db “r”, 0

buff db 0

descriptor dd -1

C.S.

afiseaza\_fisier:

mov ebx, [esp+4]

push ebx

push dword format\_nume

call [printf]

add esp, 4\*2

push dword mod\_acces

push ebx

call [fopen]

add esp, 4\*2

cmp eax, 0

je final

mov [descriptor], eax

repeta:

push dword [descriptor]

times 2 push dword 1

push dword buff

call [fread]

add esp, 4\*4

cmp eax, 0

je gata

mov edx, 0

mov dl, [buff]

push edx

push dword format\_c

call [printf]

gata:

push dword [descriptor]

call [fclose]

add esp, 4\*1

final:

ret

1. Se citeste de la tastatura numele unui fisier si apoi se citesc din fisier toate numerele existente reprezentate pe dublucuvant. Sa se afiseze numerele ai caror octeti low din cuvintele high sunt primi

DS:

format db “%s”, 0

nume\_fisier times 10 db 0

numere dd 0

format\_n db “%d”, 0

mod\_acces db “r”, 0

descriptor dd 0

CS:

push dword nume\_fisier

push dword format

call [scanf]

add esp, 4\*2

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp, 2\*4

cmp eax, 0

je final

mov [descriptor], eax

repeta:

push dword numere

push dword format\_n

push dword [descriptor]

call [fscanf]

add esp, 3\*4

cmp eax, 1

jne final\_citire

mov eax, 0

mov al, [numere+2]

mov bl, 2

prim:

div bl

cmp ah, 0

je repeta

inc bl

mov eax, 0

mov al, [numere+2]

cmp bl, al

jne prim

push dword [numere]

push dword format\_n

call [printf]

add esp 2\*4

jmp repeta

final\_citire:

push dword [descriptor]

call [fclose]

add esp, 4

final:

push dword 0

call [exit]

Ce modificari trebuie facute programului astfel incat acesta sa afiseze corect pe ecran cea mai mare valoare fara semn dintre a, b si c, indiferent de valorile a, b si c