; 1. Se da un fisier text. Sa se citeasca continutul fisierului, sa se contorizeze numarul de vocale si sa se afiseze aceasta valoare.

; Numele fisierului text este definit in segmentul de date.

bits 32

global start

extern exit, printf, scanf, fopen, fscanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fscanf msvcrt.dll

segment data use32 class=data

nume\_fisier db "test.txt",0

mod\_acces db "r",0

descriptor\_fis dd -1

len equ 100

text times len db 0

caracter db 0

format\_citire db "%c",0

contor dd 0

format db "%d",0

segment code use32 class=code

start:

deschidere:

;eax=fopen(nume\_fisier,mod\_acces)

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp,4\*2

mov [descriptor\_fis],eax

cmp eax,0

je final

repeta:

citeste:

;eax=fscanf(caracter,format\_citire,descriptor\_fis)

push dword caracter

push dword format\_citire

push dword [descriptor\_fis]

call [fscanf]

add esp, 4\*3

prelucreaza:

cmp eax,1

jne final

mov ebx,0

mov bl,[caracter]

cmp bl,'a'

je ad

cmp bl,'e'

je ad

cmp bl,'i'

je ad

cmp bl,'o'

je ad

cmp bl,'u'

je ad

jmp f

ad:

add dword[contor],1

f:

jmp repeta

final:

;fclose(descriptor\_fis)

push dword [contor]

push dword format

call [printf]

add esp,4\*2

push dword 0

call [exit]

; 2.Se da un fisier text. Sa se citeasca continutul fisierului, sa se contorizeze numarul de consoane si

; sa se afiseze aceasta valoare. Numele fisierului text este definit in segmentul de date.

bits 32

global start

extern exit, printf, scanf, fopen, fscanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fscanf msvcrt.dll

segment data use32 class=data

nume\_fisier db "test.txt",0

mod\_acces db "r",0

descriptor\_fis dd -1

len equ 100

text times len db 0

caracter db 0

format\_citire db "%c",0

contor dd 0

format db "%d",0

segment code use32 class=code

start:

deschidere:

;eax=fopen(nume\_fisier,mod\_acces)

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp,4\*2

mov [descriptor\_fis],eax

cmp eax,0

je final

repeta:

;eax=fscanf(caracter,format\_citire,descriptor\_fis)

push dword caracter

push dword format\_citire

push dword [descriptor\_fis]

call [fscanf]

add esp, 4\*3

cmp eax,1

jne final

mov edx,0

mov bl,[caracter]

cmp bl,'a'

je voc

cmp bl,'e'

je voc

cmp bl,'i'

je voc

cmp bl,'o'

je voc

cmp bl,'u'

je voc

jmp f

voc:

mov edx,1

f:

cmp edx,1

je peste

add dword[contor],1

peste:

jmp repeta

final:

;fclose(descriptor\_fis)

push dword [contor]

push dword format

call [printf]

add esp,4\*2

push dword 0

call [exit]

; 5.Se da un fisier text. Sa se citeasca continutul fisierului, sa se contorizeze numarul de caractere

; speciale si sa se afiseze aceasta valoare. Numele fisierului text este definit in segmentul de date.

bits 32

global start

extern exit, printf, scanf, fopen, fscanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fscanf msvcrt.dll

segment data use32 class=data

voc db "aeiouAEIOU"

lg equ $-voc

nume\_fisier db "test.txt",0

mod\_acces db "r",0

descriptor\_fis dd -1

len equ 100

text times len db 0

caracter db 0

format\_citire db "%c",0

contor dd 0

format db "%d",0

segment code use32 class=code

start:

;eax=fopen(nume\_fisier,mod\_acces)

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp,4\*2

mov [descriptor\_fis],eax

cmp eax,0

je final

repeta:

;eax=fscanf(caracter,format\_citire,descriptor\_fis)

push dword caracter

push dword format\_citire

push dword [descriptor\_fis]

call [fscanf]

add esp, 4\*3

cmp eax,1

jne final

mov ecx,lg

mov esi,0

mov bl,[caracter]

v:

mov al,[voc+esi]

cmp al,bl

je sari

inc esi

loop v

add dword[contor],1

sari:

jmp repeta

final:

;fclose(descriptor\_fis)

push dword [contor]

push dword format

call [printf]

add esp,4\*2

push dword 0

call [exit]

bits 32

global start

extern exit, printf, scanf, fopen, fscanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fscanf msvcrt.dll

segment data use32 class=data

voc db "abcdefghijklmnopqrstuvwxyzASDFGHJKLZXCVBNMQWERTYUIOP1234567890"

lg equ $-voc

nume\_fisier db "test.txt",0

mod\_acces db "r",0

descriptor\_fis dd -1

len equ 100

text times len db 0

caracter db 0

format\_citire db "%c",0

contor dd 0

format db "%d",0

segment code use32 class=code

start:

;eax=fopen(nume\_fisier,mod\_acces)

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp,4\*2

mov [descriptor\_fis],eax

cmp eax,0

je final

repeta:

;eax=fscanf(caracter,format\_citire,descriptor\_fis)

push dword caracter

push dword format\_citire

push dword [descriptor\_fis]

call [fscanf]

add esp, 4\*3

cmp eax,1

jne final

mov ecx,lg

mov esi,0

mov bl,[caracter]

v:

mov al,[voc+esi]

cmp al,bl

je sari

inc esi

loop v

add dword[contor],1

sari:

jmp repeta

final:

;fclose(descriptor\_fis)

push dword [contor]

push dword format

call [printf]

add esp,4\*2

push dword 0

call [exit]

; 3.Se da un fisier text. Sa se citeasca continutul fisierului, sa se contorizeze numarul de cifre pare si sa se

; afiseze aceasta valoare. Numele fisierului text este definit in segmentul de date.

bits 32

global start

extern exit, printf, scanf, fopen, fscanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fscanf msvcrt.dll

segment data use32 class=data

nume\_fisier db "test.txt",0

mod\_acces db "r",0

descriptor\_fis dd -1

len equ 100

text times len db 0

n dd 0

format\_citire db "%d",0

contor dd 0

format db "%d",0

segment code use32 class=code

start:

;eax=fopen(nume\_fisier,mod\_acces)

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp,4\*2

mov [descriptor\_fis],eax

cmp eax,0

je final

repeta:

;eax=fscanf(caracter,format\_citire,descriptor\_fis)

push dword n

push dword format\_citire

push dword [descriptor\_fis]

call [fscanf]

add esp, 4\*3

cmp eax,1

jne final

mov ebx,[n]

add ebx,0

jnp peste

add dword[contor],1

peste:

jmp repeta

final:

;fclose(descriptor\_fis)

push dword [contor]

push dword format

call [printf]

add esp,4\*2

push dword 0

call [exit]

bits 32

global start

extern exit, printf, scanf, fopen, fscanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fscanf msvcrt.dll

segment data use32 class=data

voc db "abcdefghijklmnopqrstuvwxyzASDFGHJKLZXCVBNMQWERTYUIOP1234567890"

lg equ $-voc

nume\_fisier db "test.txt",0

mod\_acces db "r",0

descriptor\_fis dd -1

len equ 100

text times len db 0

caracter db 0

format\_citire db "%c",0

contor dd 0

format db "%d",0

segment code use32 class=code

start:

;eax=fopen(nume\_fisier,mod\_acces)

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp,4\*2

mov [descriptor\_fis],eax

cmp eax,0

je final

repeta:

;eax=fscanf(caracter,format\_citire,descriptor\_fis)

push dword caracter

push dword format\_citire

push dword [descriptor\_fis]

call [fscanf]

add esp, 4\*3

cmp eax,1

jne final

mov ecx,lg

mov esi,0

mov bl,[caracter]

v:

mov al,[voc+esi]

cmp al,bl

je sari

inc esi

loop v

add dword[contor],1

sari:

jmp repeta

final:

;fclose(descriptor\_fis)

push dword [contor]

push dword format

call [printf]

add esp,4\*2

push dword 0

call [exit]

; 6.Se da un fisier text. Sa se citeasca continutul fisierului, sa se determine cifra cu cea mai mare frecventa si sa se afiseze

; acea cifra impreuna cu frecventa acesteia. Numele fisierului text este definit in segmentul de date.

bits 32

global start

extern exit, printf, scanf, fopen, fscanf

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fscanf msvcrt.dll

segment data use32 class=data

voc db "abcdefghijklmnopqrstuvwxyzASDFGHJKLZXCVBNMQWERTYUIOP1234567890"

lg equ $-voc

nume\_fisier db "test.txt",0

mod\_acces db "r",0

descriptor\_fis dd -1

len equ 100

text times len db 0

numar dd 0

format\_citire db "%d",0

contor dd 0

maxim dd 0

format db "Maxim este %d de %d",0

segment code use32 class=code

start:

;eax=fopen(nume\_fisier,mod\_acces)

push dword mod\_acces

push dword nume\_fisier

call [fopen]

add esp,4\*2

mov [descriptor\_fis],eax

cmp eax,0

je final

repeta:

;eax=fscanf(caracter,format\_citire,descriptor\_fis)

push dword numar

push dword format\_citire

push dword [descriptor\_fis]

call [fscanf]

add esp, 4\*3

cmp eax,1

jne final

mov ebx,[numar]

cmp ebx,[maxim]

jb sari

mov dword[maxim],ebx

add dword[contor],1

sari:

jmp repeta

final:

;fclose(descriptor\_fis)

push dword [contor]

push dword [maxim]

push dword format

call [printf]

add esp,4\*2

push dword 0

call [exit]

; 27.Se da un fisier text. Fisierul contine numere (in baza 10) separate prin spatii. Sa se citeasca continutul acestui fisier,

; sa se determine minimul numerelor citite si sa se scrie rezultatul la sfarsitul fisierului.

bits 32

global start

extern exit,fopen,fclose,fscanf,fprintf,printf

import exit msvcrt.dll

import fopen msvcrt.dll

import fclose msvcrt.dll

import fscanf msvcrt.dll

import printf msvcrt.dll

import fprintf msvcrt.dll

segment data use32 class=data

fisier db "lab8ex27.txt",0

mod\_acces db "a+",0

descriptor dd -1

nr dd 0

format db "%d",0

minim dd 100

fspatiu db 13,10,"%d",0

segment code use32 class=code

start:

push dword mod\_acces

push dword fisier

call [fopen]

add esp,4\*2

mov [descriptor],eax

cmp eax,0

je final

repeta:

push dword nr

push dword format

push dword [descriptor]

call [fscanf]

cmp eax,1

jne f

mov ebx,[nr]

cmp ebx,[minim]

jae peste

mov dword[minim],ebx

peste:

jmp repeta

f:

push dword [minim]

push dword fspatiu

push dword [descriptor]

call [fprintf]

add esp,4\*2

push dword[descriptor]

call [fclose]

final:

push dword 0

call [exit]

; 27.Se da un nume de fisier (definit in segmentul de date). Sa se creeze un fisier cu numele dat, apoi sa se

; citeasca de la tastatura numere si sa se scrie din valorile citite in fisier numerele divizibile cu 7, pana cand se

; citeste de la tastatura valoarea 0.

bits 32

global start

extern exit,fopen,fclose,fscanf,fprintf,printf,scanf

import exit msvcrt.dll

import fopen msvcrt.dll

import fclose msvcrt.dll

import fscanf msvcrt.dll

import scanf msvcrt.dll

import printf msvcrt.dll

import fprintf msvcrt.dll

segment data use32 class=data

fisier db "lab8ex171.txt",0

mod\_acces db "w",0

descriptor dd -1

nr dd 0

format db "%d",0

minim dd 100

fspatiu db "%d ",0

sapte dd 7

segment code use32 class=code

start:

push dword mod\_acces

push dword fisier

call [fopen]

add esp,4\*2

mov [descriptor],eax

cmp eax,0

je final

repeta:

push dword nr

push dword format

call [scanf]

mov eax,[nr]

cmp eax,0

je f

mov edx,0

div dword[sapte]

cmp edx,0

jne peste

push dword[nr]

push dword fspatiu

push dword [descriptor]

call [fprintf]

peste:

jmp repeta

f:

push dword[descriptor]

call [fclose]

final:

push dword 0

call [exit]

; 14.Se dau un nume de fisier si un text (definite in segmentul de date).

; Textul contine litere mici, litere mari, cifre si caractere speciale. Sa se transforme toate literele mari din textul

; dat in litere mici. Sa se creeze un fisier cu numele dat si sa se scrie textul obtinut in fisier.

bits 32

global start

extern exit,fopen,fclose,fscanf,fprintf,printf,scanf

import exit msvcrt.dll

import fopen msvcrt.dll

import fclose msvcrt.dll

import fscanf msvcrt.dll

import scanf msvcrt.dll

import printf msvcrt.dll

import fprintf msvcrt.dll

segment data use32 class=data

text db "Asnmb Llk !nsL"

lg equ $-text

fisier db "lab8ex14.txt",0

mod\_acces db "w",0

descriptor dd -1

nr dd 0

format db "%s",0

minim dd 100

fspatiu db "%s",0

cifre db "0123456789"

literemari db "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

nou times lg db 0

segment code use32 class=code

start:

push dword mod\_acces

push dword fisier

call [fopen]

add esp,4\*2

mov [descriptor],eax

cmp eax,0

je final

mov ecx,lg

mov esi,0

repeta1:

push ecx

mov al,[text+esi]

mov edi,0

cmp al,'A'

jb peste

cmp al,'Z'

ja peste

transf:

mov bl,'a'-'A'

add al,bl

peste:

mov [nou+esi],al

inc esi

pop ecx

loop repeta1

push dword nou

push dword fspatiu

push dword [descriptor]

call [fprintf]

add esp,4\*3

f:

push dword[descriptor]

call [fclose]

final:

push dword 0

call [exit]

; 6.Se da un fisier text. Sa se citeasca continutul fisierului, sa se determine cifra cu cea mai mare frecventa

; si sa se afiseze acea cifra impreuna cu frecventa acesteia. Numele fisierului text este definit in segmentul de date.

bits 32 ; assembling for the 32 bits architecture

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit,fopen,fclose,fscanf,printf ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

import fopen msvcrt.dll

import fclose msvcrt.dll

import fscanf msvcrt.dll

import printf msvcrt.dll

; our data is declared here (the variables needed by our program)

segment data use32 class=data

frecventa times 10 db 0

maxim db 0

fmax dd 0

cif dd 0

text db 0

fisier db "daiana.txt",0

mod\_acces db "r",0

descriptor dd -1

cifra dd 0

caracter dd 0

format db "%c",0

form db "Cifra %d apare de %d ori",0

; our code starts here

segment code use32 class=code

start:

push dword mod\_acces

push dword fisier

call [fopen]

add esp,4\*2

cmp eax,0

je final

mov [descriptor],eax

citire:

push dword caracter

push dword format

push dword [descriptor]

call [fscanf]

add esp,4\*3

cmp eax,1

jne f

mov al,byte[caracter]

cmp al,'0'

jb citire

cmp al,'9'

ja citire

sub al,'0'

add byte[frecventa+eax],1

jmp citire

f:

mov ecx,11

mov esi,0

r:

mov al,[frecventa+esi]

cmp al,[maxim]

jbe peste

mov byte[maxim],al

mov dword[cif],esi

peste:

inc esi

loop r

mov eax,0

mov al,[maxim]

mov dword[fmax],eax

push dword[fmax]

push dword[cif]

push dword form

call [printf]

add esp,4\*2

push dword[descriptor]

call [fclose]

final:

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program

; 21. Se dau un nume de fisier si un text (definite in segmentul de date). Textul contine litere mici, cifre si spatii.

; Sa se inlocuiasca toate cifrele de pe

; pozitii impare cu caracterul ‘X’. Sa se creeze un fisier cu numele dat si sa se scrie textul obtinut in fisier.

bits 32 ; assembling for the 32 bits architecture

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit,fopen,fclose,fscanf,fprintf ; tell nasm that exit exists even if we won't be defining it

import exit msvcrt.dll ; exit is a function that ends the calling process. It is defined in msvcrt.dll

import fopen msvcrt.dll

import fclose msvcrt.dll

import fscanf msvcrt.dll

import fprintf msvcrt.dll

; our data is declared here (the variables needed by our program)

segment data use32 class=data

fisier db "luni.txt",0

mod\_acces db "w",0

descriptor dd -1

format db "%s",0

x db 'X'

text db "a1 b2"

lg equ $-text

; our code starts here

segment code use32 class=code

start:

mov esi,0

mov ecx,lg

repeta:

mov al,[text+esi]

cmp al,'0'

jb peste

cmp al,'9'

ja peste

mov ebx,esi

add ebx,1

shr ebx,1

jnc peste

mov bl,[x]

mov [text+esi],bl

peste:

inc esi

loop repeta

push dword mod\_acces

push dword fisier

call [fopen]

add esp,4\*2

cmp eax,0

je final

mov [descriptor],eax

push dword text

push dword format

push dword [descriptor]

call [fprintf]

add esp,4\*3

push dword[descriptor]

call [fclose]

add esp,4\*1

final:

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program