assume ds:data,cs:code

data segment

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ax,4C00h

int 21h

code ends

end start

;1.Sa se tipareasca continutul registrilor si suma primilor 4 biti ai lor.

assume ds:data,cs:code

data segment

max db 'Continutul registrului ax este=$'

sax db 'Suma primilor 4 biti ai registrului ax este=$'

creg db 18 dup(?)

sreg db ?

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ax,1001000111111111b

mov si,-1

mov cx,16

mov sreg,0

bucla:

shl ax,1

inc si

jc unu

mov creg[si],'0'

jmp cont

unu:

mov creg[si],'1'

cmp si,11

jbe cont

add sreg,1

cont:

loop bucla

inc si

mov creg[si],10 ;pun in sir caract.10(Enter)ca sa treaca pe linia urm

inc si

mov creg[si],'$' ;pun $. Indica sfarsitul sirului de tiparit

mov dx,offset max ;incarc in dx mesaju

mov ah,09h ;incarc functia 09h, care tipareste un string

int 21h ;tiparesc mesajul

mov dx,offset creg ;incarc in dx continutul lui ax

mov ah,09h

int 21h ;il tiparesc

mov dx, offset sax ;incarc mesajul urmator (ala cu suma celor...)

mov ah,09h

int 21h ;il tiparesc

mov si,0

mov ah,sreg ;pun in ah suma celor patru biti

add ah,48 ;ii adun 48.Rezulta suma celor patru biti ca si caracter

;ah=4; ah+48=52; 52 este caracterul 4

mov creg[si],ah

inc si

mov creg[si],10 ;pun si enter la sfarsitul liniei

inc si

mov creg[si],'$' ;si caracterul ala

mov dx,offset creg

mov ah,09h

int 21h

mov ax,4C00h

int 21h

code ends

end start

;2. Sa se citeasca cite un caracter de la tastatura, fara ecou. Daca

; este cifra, sa se tipareasca pe ecran, daca este caracterul

; '$', se termina programul, in orice alta situatie se adauga

; caracterul intr-un buffer care se va tipari in final pe ecran.

assume cs:code

assume ds:data

data segment

buffer db 20 dup(?)

data ends

code segment

start:

mov ax,data

mov ds,ax

mov si,0

mov ah,08h ; console char input without echo - adica citeste un caracter cand ;apelez intreruperea...

int 21h ;citesc caracter pe care il pune in al

bucla:

cmp al,'$'

je sfbucla ;daca e '$' am terminaT

cmp al,'0' ;DACA E MAI MARE DECAT 0 (VERIFIC DACA E NUMAR)

jb et

cmp al,'9' ;daca e mai mic decat 9 am numar

ja et

mov dl,al ;cand ajung aici am numar, pregatesc tiparirea

mov ah,02h ;pun in ah 02h adica character output

int 21h ;tiparesc caracterul din dl

jmp et1 ;ma duc la et1 (citirea urm caracter)

et:

mov buffer[si],al ;daca ajung aici am alt carcater

add si,1 ;il aDAUG IN BUFFER si cresc adresa curenta din buffer

et1:

mov ah,08h ;pregatesc citirea

int 21h ;citesc un caracter

jmp bucla ;repet procedeul

sfbucla:

mov buffer[si],al ;mut in buffer ultimul caracter ($)

mov ah,09h ;09h - print string pregateste tiparirea

mov dx,offset buffer ;dx offsetul bufferului

int 21h ;tiparesc bufferul

mov ax,4C00h ;iesire din program...

int 21h

code ends

end start

;Daca unitatea implicita este cea citita de la tastatura (X),

;atunci sa se selecteze ca si unitate implicita unitatea A:,

;in caz contrar, sa se selecteze unitatea X:. Orice situatie de

;eroare va fi semnalata.

assume ds:data,cs:code

data segment

unitate\_cit db ?

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,08h ;citesc o unitate

int 21h

cmp al,'a'

je ea

cmp al,'c'

je ec

cmp al,'d'

je ed

ea:

mov unitate\_cit,0

jmp maideparte

ec:

mov unitate\_cit,2

jmp maideparte

ed:

mov unitate\_cit,3

jmp maideparte

maideparte:

mov ah,19h ;al=unitatea implicita-driver-ul curent

int 21h

cmp al,unitate\_cit

je unit\_a

jmp unit\_x

unit\_a:

mov ah,0eh ;seteaza driver-ul curent

mov dl,0 ;selecteaza unitatea 0=A

int 21h

jmp sfarsit

unit\_x:

mov ah,0eh

mov dl,unitate\_cit ;selecteaza unitatea unitate\_cit=X

int 21h

jmp sfarsit

sfarsit:

mov ax,4c00h

int 21h

code ends

end start

;4. Sa se afiseze data curenta si ziua curenta din saptamina (in litere).

assume cs:code

assume ds:data

data segment

lun db ' luni $'

ma db ' marti $'

mi db ' miercuri $'

joi db ' joi $'

vi db ' vineri $'

sa db ' sambata $'

du db ' duminica $' ;zilele din saptamana...

zi db 2 dup (?),'-$'

lu db 2 dup (?),'-$'

an db 4 dup (?),' $'

aux db 10 ;var auxiliara folosita la calcularea datei..

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,2Ah ;pregatesc citirea datei din sistem

int 21h ;citesc data din sistem.voi avea:ax 0..6 ziua (0-duminica), cx-anul, ;dh-luna, dl-ziua(1-31)

mov bx,ax ;copiez registrul ax in bx...adica tin minte codul zilei

mov ax,cx ;prima data anul.

div aux ;impart la 10

add ah,'0' ;restul impartirii(ah) va fi ultima cifra a anului si va fi caracter

mov an[3],ah ;pun cifrele in sirul an in ordine inversa folosind ah

mov ah,0 ;golesc restul si mai impart odata, obtin penultima cifra in ah

div aux

add ah,'0'

mov an[2],ah ;si tot asa...

mov ah,0

div aux

add ah,'0'

mov an[1],ah

mov ah,0

div aux

add ah,'0'

add an[0],ah

mov al,dh ;copiez luna in al

xor ah,ah ;in ah pun 0, deci in ax am luna

div aux ;impart la 10, voi avea in ah ultima cifra si in al prima cifra

add ah,'0'

mov lu[1],ah

add al,'0'

mov lu[0],al ;am mutat luna in sirul lu

mov al,dl ;copiez ziua in al

xor ah,ah ;pun in ah 0

div aux

add ah,'0' ;dupa impartire am in sh ultima cifra si in al prima cifra

mov zi[1],ah ;mut ziua in sirul zi

add al,'0'

mov zi[0],al

mov ah,09h ;mut in al 09h (09hrint string), adica pregatesc tiparirea

mov dx,offset zi ;tiparesc ziua

int 21h

mov dx,offset lu ;tiparesc luna

int 21h

mov dx,offset an ;tiparesc anul

int 21h

; afisam ziua din saptamana

; codul zilei a fost in al si l-am mutat in bl

mov ah,09h

cmp bl,0 ;fac 7 comparatii cate una pentru fiecare zi si afisez ziua ;corespunzatoare.cred ca ar fi MULT mai simplu sa mut in ds:si offsetul sirului ;corespunzator zilei si sa tiparesc folosind 09h si in loc de offset sa pun ;direct valoarea din al ar trebui declarate variabilele corespunzatoare zilelor ;incepand cu duminica da mi leeeennnneeeeeeee amuuuuu

ja et1

mov dx,offset du

int 21h

jmp final

et1:

cmp bl,1

ja et2

mov dx,offset lun

int 21h

jmp final

et2:

cmp bl,2

ja et3

mov dx,offset ma

int 21h

jmp final

et3:

cmp bl,3

ja et4

mov dx,offset mi

int 21h

jmp final

et4:

cmp bl,4

ja et5

mov dx,offset joi

int 21h

jmp final

et5:

cmp bl,5

ja et6

mov dx,offset vi

int 21h

jmp final

et6:

mov dx,offset sa

int 21h

final:

mov ax,4C00h ;shi pa.

int 21h

code ends

end start

;6. Sa se afiseze ora sistem curenta sub forma hh:mm si sa se specifice

;cu sirul 'AM' sau 'PM' daca este antemeridian sau postmeridian.

assume cs:code,ds:data

data segment

am db ' AM $'

pm db ' PM $'

hh db 2 dup (?),':$'

mm db 2 dup (?),' $'

t db ?

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,2Ch ;intreruperea da ora sistemului ch->ora

; cl->minute

; dh->secunde

; dl->miimi de secunda

int 21h

mov al,ch;punem ora

mov ah,0

mov bl,12;vedem daca e dimineata sau seara

div bl

mov t,al ;catul in al

;luam ora si transformam in sir pentru a putea afisa

mov al,ch

mov bl,10

mov ah,0

div bl

add ah,'0'

mov hh[1],ah

add al,'0'

mov hh[0],al

;luam minutele si le transformam in sir pentru a le putea afisa

mov al,cl ;cl-minute

mov ah,0

mov bl,10

div bl

add ah,'0'

add al,'0'

mov mm[0],al

mov mm[1],ah

mov ah,09h

mov dx,offset hh

int 21h

mov dx,offset mm

int 21h

cmp t,0

jne seara

mov dx,offset am

mov ah,09h

int 21h

jmp final

seara:

mov dx,offset pm

mov ah,09h

int 21h

final:

final:

mov ax,4C00h

int 21h

code ends

end start

; Sa se afiseze spatiul liber de pe o discheta

assume cs:code,ds:data

data segment

mesaj db 'Spatiul liber de pe discheta este:',13,10,' $'

sir db 4 dup(?),' $'

ori dw 1024

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,09h

mov dx,offset mesaj

int 21h

mov ah,36h ;apelez intreruperea pt obtinerea spatiului liber de pe A

mov dl,1

int 21h

mov ax,bx

mul cx

div ori ;am obtinut in al catul si in ah restul

;tb sa transform in sir

mov si,4

mov cx,10

again:

mov dx,0

div cx

add dl,'0'

mov sir[si],dl

dec si

cmp ax,0

jne again

mov ah,09h

mov dx,offset sir

int 21h

mov ax,4C00h

int 21h

code ends

end start

;9. Sa se afiseze spatiul ocupat pe disc.

assume ds:data

assume cs:code

data segment

text db ' spatiu ocupat : $'

nr db 6 dup(?),10,13,' $'

z dw 10

nrbit dw 1024

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,36h

mov dl,1

int 21h

sub dx,bx

mov ax,dx

mul cx

div nrbit

mov si,4

bucla:

mov dx,0

div z

add dl,'0'

mov nr[si],dl

cmp ax,0

dec si

je sfbucla

jmp bucla

sfbucla:

mov ah,09h

mov dx,offset text

int 21h

mov dx,offset nr

int 21h

mov ax,4C00h

int 21h

code ends

end start

;Sa se stearga un fisier al carui nume va fi introdus de la

;tastatura,orice situatie de eroare va fi semnalata printr-un mesaj

;corespunzator

assume ds:data,cs:code

data segment

m1 db 'Nu s-a gasit fisierul',10,13,'$'

m2 db 'Nu s-a gasit calea',13,10,'$'

m3 db 'Acces interzis',13,10,'$'

m4 db 'Fisier sters',13,10,'$'

msg db 'Numele fisierului: ',13,10,'$'

maxnumefis db 12

lnumefis db ?

numefis db 12 dup (?)

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah, 09h ;afiseaza mesajul 'Numele fisierului:'

;ah-se lucreaza cu intreruperile

mov dx, offset msg;adresa de inceput a mesajului

int 21h

mov ah,0ah ;citeste numele fisierului

mov dx,offset maxnumefis;pun in dx lg. maxima pe care o poate avea sirul

int 21h

;In urma citirii la adresa maxnumefis+2=numefis se memoreaza numele fisierului

;citit la adresa maxnumefis+1=lnumefis se memoreaza dimensiunea sirului de

;caractere care reprezinta numele fisierului transformam numele fisierului

;intr-un sir ASCII

mov al,lnumefis

xor ah,ah ;se face 0 in ah

mov si, ax

mov numefis[si],0 ;transform(prin punerea caracterului 0 la urma)sirul

; in string

mov ah,41h ;sterge fisierul

lea dx,numefis;

int 21h

;tratam erorile, afisand mesaje corespunzatoare

cmp ax,2

je eroare1 ;nu gaseste fisierul

cmp ax,3

je eroare2 ;nu gaseste calea

cmp ax,5

je eroare3 ;acces interzis

lea dx,m4

mov ah,09h

int 21h

jmp sf

eroare1:

lea dx,m1

mov ah,09h

int 21h

jmp sf

eroare2:

lea dx,m2

mov ah,09h

int 21h

jmp sf

eroare3:

lea dx,m3

mov ah,09h

int 21h

sf:

mov ax,4c00h

int 21h

code ends

end start

SAU

assume cs:code,ds:data

data segment

mesaj1 db 'Dati numele fisierului ',13,10,'$'

mesaj2 db 'Fisierul a fost sters ',13,10,'$'

eroare1 db 'Nu s-a gasit fisierul ',13,10,'$'

eroare2 db 'Nu s-a gasit calea ',13,10,'$'

eroare3 db 'Acces interzis ',13,10,'$'

numefis db 12 dup(?)

lungmax db 12

lung db ?

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,09h ;afisez mesaj1 cu nume fisier

mov dx,offset mesaj1 ;

int 21h ;

mov ah,0ah ;citesc numele fisierului

mov dx,offset lungmax ;

int 21h ;

mov al,lung ;transform numefis in ascii

mov ah,0 ;

mov si,ax ;

mov numefis[si],0 ;

mov ah,41h ;sterg fis.,codurile erorilor se returneaza in ax

lea dx,numefis ;

int 21h ;

cmp ax,2

je err1

cmp ax,3

je err2

cmp ax,5

je err3

mov ah,09h

lea dx,mesaj2

int 21h

jmp sfarsit

err1:

lea dx,eroare1

mov ah,09h

int 21h

jmp sfarsit

err2:

lea dx,eroare2

mov ah,09h

int 21h

jmp sfarsit

err3:

lea dx,eroare3

mov ah,09h

int 21h

sfarsit:

mov ax,4c00h

int 21h

code ends

end start

; 9.Sa se afiseze numele si continutul directorului curent.

assume cs:code,ds:data

data segment

nume db 64 dup(?),'$'

fis db '\*.\*'

fnf db 'File not found.$'

pnf db 'Path not found.$'

nmf db 'No more files to be found.$'

buffer db 43 dup(?),'$'

fisier db 13 dup(?),'$'

data ends

code segment

start:

mov ax,data

mov ds,ax

; returneaza numele directorului curent

mov ah,47h ; dl->numarului driver-ului

mov dl,0 ; in dl se pune 0 adica nu se stie driver-ul curent

mov si,offset nume ;in ds:si->numele directorului curent

int 21h

mov ah,09h

mov dx,offset nume

int 21h

mov fis[3],0 ;fac sir

mov ah,1ah ; 1Ah seteaza DTA (disk transfer adress)

mov dx,offset buffer

int 21h

mov ah,4Eh ; gaseste primul fisier din direstorul curent

mov dx,offset fis

mov cx,27h ;

int 21h

jc eroare

mov si,0

mov cx,13

mov di,30

iar:

mov bl,buffer[di]

mov fisier[si],bl ; la offset-ul 1Eh se pune numele si extensia fisierului

inc si

inc di

loop iar

mov si,0

mov ah,09h

mov dx,offset fisier

int 21h

bucla:

mov ah,1Ah

mov dx,offset buffer

mov ah,4Fh

mov dx,offset fis

mov cx,27h

int 21h

jc nomorefiles

mov si,0

mov cx,13

mov di,30

iar1:

mov bl,buffer[di]

mov fisier[si],bl

inc si

inc di

loop iar1

mov ah,09h

mov dx,offset fisier

int 21h

je sf

jmp bucla

eroare:

cmp ax,2

je filenotfound

cmp ax,3

je pathnotfound

cmp ax,18

je nomorefiles

jmp sf

filenotfound:

mov ah,09h

mov dx,offset fnf

int 21h

jmp sf

pathnotfound:

mov ah,09h

mov dx,offset pnf

int 21h

jmp sf

nomorefiles:

mov ah,09h

mov dx,offset nmf

int 21h

sf:

mov ax,4C00h

int 21h

code ends

end start

;Sa se afiseze atributele unui fisier al carui nume se da de la tastatura

assume ds:data, cs:code

data segment

CR equ 13

LF equ 10

maxFile db 13

lenFile db ?

strFile db 13 dup('$')

msgInput db 'Introduceti numele fisierului: $'

msgOutput db 'Atributele fisierului sunt:', CR, LF, '$'

aReadOnly db ' [x] Read-only', CR, LF, '$'

aHidden db ' [x] Hidden', CR, LF, '$'

aSystem db ' [x] System', CR, LF, '$'

aArchive db ' [x] Archive', CR, LF, '$'

nReadOnly db ' [ ] Read-only', CR, LF, '$'

nHidden db ' [ ] Hidden', CR, LF, '$'

nSystem db ' [ ] System', CR, LF, '$'

nArchive db ' [ ] Archive', CR, LF, '$'

msgError db 'Eroare la accesarea atributelui fisierului (fisierul nu exista?).$', CR, LF

data ends

code segment

start:

mov ax, data ; ax &lt;- adresa segmentului de date

mov ds, ax ; ds &lt;- dseg

mov es, ax ; es &lt;- dseg

; Afisez un mesaj pt introducerea numelui fisierului

mov dx, offset msgInput

mov ah, 09h

int 21h

; Citesc numele fisierului

mov dx, offset maxFile ; pointer spre input buffer

mov ah, 0Ah ; buffered input

int 21h ; apelez intreruperea

; Las un rand liber

mov dl, LF ;

mov ah, 02h ; char output

int 21h

; Afisez un mesaj

mov dx, offset msgOutput

mov ah, 09h

int 21h

; Transform numele fisierului intr-un sir ASCIIZ

mov al, lenFile

xor ah, ah

mov si, ax

mov [strFile+si], 0

; Apelez functia get/set attributes

mov dx, offset strFile

mov ax, 4300h

int 21h

jc errFisier

; Verific atributele fisierului

test cx, 1h

jz notReadOnly

mov dx, offset aReadOnly

mov ah, 09h

int 21h

jmp testHidden

notReadOnly:

mov dx, offset nReadOnly

mov ah, 09h

int 21h

testHidden:

test cx, 2h

jz notHidden

mov dx, offset aHidden

mov ah, 09h

int 21h

jmp testSystem

notHidden:

mov dx, offset nHidden

mov ah, 09h

int 21h

testSystem:

test cx, 4h

jz notSystem

mov dx, offset aSystem

mov ah, 09h

int 21h

jmp testArchive

notSystem:

mov dx, offset nSystem

mov ah, 09h

int 21h

testArchive:

test cx, 20h

jz notArchive

mov dx, offset aArchive

mov ah, 09h

int 21h

jmp sfarsit

notArchive:

mov dx, offset nArchive

mov ah, 09h

int 21h

jmp sfarsit

errFisier:

mov dx, offset [msgError]

mov ah, 09h

int 21h

sfarsit:

mov ax, 4C00h

int 21h

code ends

end start

; 11. Sa se pozitioneze atributele unui fisier al carui nume va fi introdus de

; la tastatura. Noile atribute vor fi determinate in urma unui dialog cu

; utilizatorul de la tastatura.

assume cs:code,ds:data

data segment

file db 10 dup(?),'$'

mesaj db 10,13,'Dati numele fisierului: $',10,13

fnf db 10,13,'File not found. $',10,13

pnf db 10,13,'Path not found. $',10,13

acd db 10,13,'Acces denied. $',10,13

normal db 10,13,'Normal? (y/n)$',10,13

rdonly db 10,13,'Read-only? (y/n)$',10,13

hidden db 10,13,'Hidden? (y/n)$',10,13

system db 10,13,'System? (y/n)$',10,13

archive db 10,13,'Archive? (y/n)$',10,13

data ends

code segment

start:

mov ax,data

mov ds,ax

mov si,0

mov ah,09h

mov dx,offset mesaj

int 21h

mov ah,01h;citesc un caracter cu ecou

int 21h

bucla:

mov file[si],al ;citesc numele fisierului

inc si

int 21h

cmp al,13

je sf1

jmp bucla

sf1:

mov file[si],0;il transforma in ascii

mov ah,09h

mov dx,offset normal

int 21h

mov ah,01h

int 21h

cmp al,'y'

je n

et1:

mov ah,09h

mov dx,offset rdonly

int 21h

mov ah,01h

int 21h

cmp al,'y'

je r

et2:

mov ah,09h

mov dx,offset hidden

int 21h

mov ah,01h

int 21h

cmp al,'y'

je h

et3:

mov ah,09h

mov dx,offset system

int 21h

mov ah,01h

int 21h

cmp al,'y'

je s

et4:

mov ah,09h

mov dx,offset archive

int 21h

mov ah,01h

int 21h

cmp al,'y'

je a

jmp sf

n:

mov ah,43h

mov al,01h

mov dx,offset file

mov cx,00h

int 21h

jc eroare

jmp et1

r:

mov ah,43h;seteaza atributele fisierului

mov al,01h

mov dx,offset file

mov cx,01h

int 21h

jc eroare

jmp et2

h:

mov ah,43h

mov al,01h

mov dx,offset file

mov cx,02h

int 21h

jc eroare

jmp et3

s:

mov ah,43h

mov al,01h

mov dx,offset file

mov cx,04h

int 21h

jc eroare

jmp et3

a:

mov ah,43h

mov al,01h

mov dx,offset file

mov cx,20h

int 21h

jc eroare

jmp sf

eroare:

cmp ax,2

je filenotfound

cmp ax,3

je pathnotfound

cmp ax,5

je accesdenied

filenotfound:

mov ah,09h

mov dx,offset fnf

int 21h

jmp sf

pathnotfound:

mov ah,09h

mov dx,offset pnf

int 21h

jmp sf

accesdenied:

mov ah,09h

mov dx,offset acd

int 21h

jmp sf

sf:

mov ax,4C00h

int 21h

code ends

end start

; 12. Sa se citeasca de la tastatura doua nume de fisiere. Sa se copieze

; primul fisier in cel de-al doilea. Se va semnala orice situatie de eroare.

assume cs:code,ds:data

data segment

file1 db 10 dup(?)

file2 db 10 dup(?)

f1nf db 'File 1 not found.',13,10,'$'

f2nf db 'File 2 not found.',13,10,'$'

p1nf db 'Path 1 not found.',13,10,'$'

p2nf db 'Path 2 not found.',13,10,'$'

ac1d db 'Acces to file 1 denied.',13,10,'$'

ac2d db 'Acces to file 2 denied.',13,10,'$'

buffer db 30 dup(?)

mesaj1 db 'Dati numele primului fisier: ',13,10,'$'

mesaj2 db 'Dati numele celui de-al doilea fisier: ',13,10,'$'

nbr dw ?

nbw dw ?

data ends

code segment

start:

mov ax,data

mov ds,ax

mov si,0

mov ah,09h

mov dx,offset mesaj1

int 21h

mov ah,01h

int 21h

bucla1:

mov file1[si],al

inc si

int 21h

cmp al,13

je sf1

jmp bucla1

sf1:

mov file1[si],0

mov si,0

mov ah,09h

mov dx,offset mesaj2

int 21h

mov ah,01h

int 21h

bucla2:

mov file2[si],al

inc si

int 21h

cmp al,13

je sf2

jmp bucla2

sf2:

mov file2[si],0

; deschidem fisierul 1 pentru citire

mov ah,3dh

mov al,0;cu 0 accesam pt citire

mov dx,offset file1

int 21h

jc openerror1

mov bx,ax ; salvam identificatorul fisierului in bx

; citim din fisier si salvam in buffer

goon1:

mov ah,3fh

mov cx,30 ;citim maxim 30 de caractere

mov dx,offset buffer

int 21h

jc readerror

;mov nbr,ax ;nbr=number of bytes read

;cmp nbr,30

;je goon1

; deschidem fisierul 2 pt scriere

mov ah,3dh

mov al,00000001b

mov dx,offset file2

int 21h

jc openerror2

mov bx,ax ; salvam identificatorul fisierului in bx

; scriem in fisier ce avem in buffer

goon2:

mov ah,40h

mov cx,30 ;scriem maxim 30 de caractere

mov dx,offset buffer

int 21h

jc writeerror

;mov nbw,ax ;nbw=number of bytes written

;cmp nbw,30

;je goon2

jmp sf

openerror1:

cmp ax,2

je file1nf

cmp ax,2

je path1nf

cmp ax,5

je acces1d

jmp sf

openerror2:

cmp ax,2

je file2nf

cmp ax,2

je path2nf

cmp ax,5

je acces2d

jmp sf

readerror:

cmp ax,5

je acces1d

jmp sf

writeerror:

cmp ax,5

je acces2d

jmp sf

file1nf:

mov ah,09h

mov dx,offset f1nf

int 21h

jmp sf

file2nf:

mov ah,09h

mov dx,offset f2nf

int 21h

jmp sf

path1nf:

mov ah,09h

mov dx,offset p1nf

int 21h

jmp sf

path2nf:

mov ah,09h

mov dx,offset p2nf

int 21h

jmp sf

acces1d:

mov ah,09h

mov dx,offset ac1d

int 21h

jmp sf

acces2d:

mov ah,09h

mov dx,offset ac2d

int 21h

jmp sf

sf:

mov ax,4C00h

int 21h

code ends

end start

; 13.Sa se citeasca de la tastatura un nume de fisier si un nume de director.

; Sa se afiseze un mesaj corespunzator daca fisierul exista sau nu in

; directorul dat.

assume cs:code,ds:data

data segment

mesaj1 db 10,13,'Dati numele fisierului: $',10,13

mesaj2 db 10,13,'Dati numele directorului: $',10,13

file db 30 dup(?)

dir db 30 dup(?)

dnf db 10,13,'Directory not found.$',10,13

found db 10,13,'The file was found.$',10,13

fnf db 10,13,'File not found.$',10,13

a dw ?

data ends

code segment

start:

mov ax,data

mov ds,ax

mov si,0

mov ah,09h

mov dx,offset mesaj1

int 21h

mov ah,01h

int 21h

bucla1:

mov file[si],al

inc si

int 21h

cmp al,13

je sf1

jmp bucla1

sf1:

mov file[si],0

mov si,0

mov ah,09h

mov dx,offset mesaj2

int 21h

mov ah,01h

int 21h

bucla2:

mov dir[si],al

inc si

int 21h

cmp al,13

je sf2

jmp bucla2

sf2:

mov dir[si],0

;schimbam directorul in cel pe care l-am citit

mov ah,3Bh

mov dx,offset dir

int 21h

jc dirnotfound

;cautam fisierul

mov ah,4Eh

mov cx,27h

mov dx,offset file

int 21h

mov a,ax ; a=codul de eroare

jc eroare

mov ah,09h

mov dx,offset found

int 21h

jmp sf

dirnotfound:

mov ah,09h

mov dx,offset dnf

int 21h

jmp sf

eroare:

cmp a,2

je filenotfound

cmp a,3

je filenotfound

cmp a,18

je filenotfound

jmp sf

filenotfound:

mov ah,09h

mov dx,offset fnf

int 21h

sf:

mov ax,4C00h

int 21h

code ends

end start

SAU

; 13.Sa se citeasca de la tastatura un nume de fisier si un nume de director.

; Sa se afiseze un mesaj corespunzator daca fisierul exista sau nu in

; directorul dat.

assume cs:code,ds:data

data segment

mesajfis db 'Dati numele fisierului:',13,10,'$'

mesajdir db 'Dati numele directorului:',13,10,'$'

eroare1 db 'Fisierul nu a fost gasit',13,10,'$'

eroare2 db 'Directorul nu a fost gasit',13,10,'$'

eroare3 db 'Fisierul a fost gasit',13,10,'$'

lgfismax db 30

lgfis db ?

fis db 30 dup(?),'$'

lgdirmax db 50

lgdir db ?

dir db 50 dup(?),'$'

a dw ?

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,09h

mov dx,offset mesajdir

int 21h

mov ah,0ah

mov dx,offset lgdirmax

int 21h

mov al,lgdir

mov ah,0

mov si,ax

mov dir[si],0 ;am format nume dir in ascii

mov ah,09h

mov dx,offset mesajfis

int 21h

mov ah,0ah

mov dx,offset lgfismax

int 21h

mov al,lgfis

mov ah,0

mov si,ax

mov fis[si],0 ;am format nume fis in ascii

mov ah,3bh ;ma pozitionez in directorul respectiv

mov dx,offset dir

int 21h

jc errdir

mov ah,4eh ;verific daca am fisierul in director

mov cx,27h

mov dx,offset fis

int 21h

jc errfis

mov ah,09h

mov dx,offset eroare3

int 21h

jmp sf

errfis:

mov ah,09h

mov dx,offset eroare1

int 21h

jmp sf

errdir:

mov ah,09h

mov dx,offset eroare2

int 21h

sf:

mov ax,4C00h

int 21h

code ends

end start

; 14. Sa se citeasca de la tastatura un cuvant si un nume de fisier. Sa se

; afiseze daca cuvantul exista sau nu in fisierul dat.

assume cs:code,ds:data

data segment

numefis db 12 dup(?)

cuvant db 20 dup(?)

lungfismax db 12

lungcuvmax db 20

lungfis db ?

lungcuv db ?

mcuvant db 'Dati cuvantul=',13,10,'$'

mfisier db 'Dati numele fisierului=',13,10,'$'

gasit db 'Cuvantul a fost gasit=',13,10,'$'

nugasit db 'Cuvantul nu a fost gasit=',13,10,'$'

mesaj1 db 'Fisierul nu a fost gasit ',13,10,'$'

mesaj2 db 'Nu s-a gasit calea ',13,10,'$'

mesaj3 db 'Acces interzis ',13,10,'$'

buffer db 50 dup (?)

data ends

code segment

start:

mov ax,data

mov ds,ax

mov ah,09h

mov dx,offset mfisier

int 21h

mov ah,0ah

mov dx,offset lungfismax

int 21h

mov ah,09h

mov dx,offset mcuvant

int 21h

mov ah,0ah

mov dx,offset lungcuvmax

int 21h

mov al,lungfis

mov ah,0

mov si,ax

mov numefis[si],0

mov al,lungcuv

mov ah,0

mov si,ax

mov cuvant[si],0

mov ah,3dh ;am deschis fisier pt citire

mov al,0

mov dx,offset numefis

int 21h

jc erori

;formez bufferul cu continutul fisierului

mov ah,3fh

mov cx,50

mov dx,offset buffer

jc errbuf

mov bx,ax

int 21h

;verific daca cuvantul e in buffer

mov si,0

mov di,0

mov cl,lungfis

bucla:

mov ah,cuvant[si]

cmp ah,buffer[di]

je bucla\_egal

inc di

cmp cl,di

je nueste

jmp bucla

bucla\_egal:

inc si

inc di

cmp si,lungcuv

je este

jmp bucla

nueste:

mov ah,09h

mov dx,offset nugasit

int 21h

jmp sfarsit

este:

mov ah,09h

mov dx,offset gasit

int 21h

jmp sfarsit

erori:

jmp sfarsit

errbuf:

jmp sfarsit

sfarsit:

mov ax,4c00h

int 21h

code ends

end start

; 15. Sa se citeasca de la tastatura doua nume de directoare dir1 si dir2. Sa

; se creeze directorul dir1\dir2.

assume cs:code,ds:data

data segment

mesaj1 db 10,13,'Dati numele primului director: $',10,13

mesaj2 db 10,13,'Dati numele celui de-al doilea director: $',10,13

dir1 db 30 dup(?)

dir2 db 30 dup(?)

pnf db 10,13,'Path not found.$',10,13

acd db 10,13,'Acces denied, or pathname already exists.$',10,13

gata db 10,13,'S-a creat directorul.$',10,13

data ends

code segment

start:

mov ax,data

mov ds,ax

;citim primul director

mov ah,09h

mov dx,offset mesaj1

int 21h

mov si,0

mov ah,01h

int 21h

bucla1:

mov dir1[si],al

inc si

int 21h

cmp al,13

je sf1

mov dir1[si],al

jmp bucla1

sf1:

mov dir1[si],0

;citim numele celui de-al doilea fisier

mov ah,09h

mov dx,offset mesaj2

int 21h

mov si,0

mov ah,01h

int 21h

bucla2:

mov dir2[si],al

inc si

int 21h

cmp al,13

je sf2

mov dir2[si],al

jmp bucla2

sf2:

mov dir2[si],0

mov ah,3Bh ;schimb directorul curent in dir1

mov dx,offset dir1

int 21h

jc eroare

mov ah,39h ;creaza un subdirector

mov dx,offset dir2

int 21h

jc er

mov ah,09h

mov dx,offset gata

int 21h

eroare:

cmp ax,3

je pathnotfound

jmp sf

pathnotfound:

mov ah,09h

mov dx,offset pnf

int 21h

jmp sf

er:

cmp ax,3

je pathnotfound

cmp ax,5

je accesdenied

accesdenied:

mov ah,09h

mov dx,offset acd

int 21h

sf:

mov ax,4C00h

int 21h

code ends

end start