Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 8

по дисциплине ПнаЯА

Вариант 1

Выполнил студент гр. 150502: Снитко Д.А.

Проверил: Туровец Н.О.

Минск 2022

Теоретические сведения:

1. Вектора прерываний.

2. Аппаратные прерывания.

3. Программные прерывания.

4. Инсталлятор резидентной программы.

5. Обработчик резидентной программы.

Задание:

Написать программу вывода строк текста из файла с обработчиком прерывания Ctrl-Break (Ctrl-C). При нажатии Ctrl-C программа должна приостановить вывод данных на экран (до нажатия любой клавиши).

Код:

.model tiny

.code

org 100h

START:

mov ax, cs

mov ds, ax

mov es, ax

mov bx, 80h

cmp [bx], 0

je WRONG\_PARAMETR

call GET\_FILE\_NAME

call IS\_TXT\_FILE

call INSTALL\_NEW\_HANDLER

call OPEN\_FILE

call CLEAR

call SHOW\_MESSAGE

for0:

call READ\_FROM\_FILE

call OUTPUT\_BUFFER

jmp for0

EXIT:

call INSTALL\_OLD\_HANDLER

call CLOSE\_FILE

mov ah, 4ch

int 21h

WRONG\_PARAMETR:

lea dx, PARAMETR\_ERROR

mov ah, 9

int 21h

mov ah, 4ch

int 21h

;SHOW MESSAGE;

SHOW\_MESSAGE proc

mov ax, 0b800h

mov es, ax

lea di, MESSAGE\_TO\_EXIT

xor bx, bx

mov dh, 00000100b

MESSAGE\_CYCLE:

mov dl, [di]

mov es:[bx], dx

add bx, 2

inc di

cmp [di], '$'

jne MESSAGE\_CYCLE

mov dl, ' '

FILL\_STRING\_CYCLE:

mov es:[bx], dx

add bx, 2

cmp bx, 162

jne FILL\_STRING\_CYCLE

ret

SHOW\_MESSAGE endp

;GET FILE NAME;

GET\_FILE\_NAME proc

mov di, 81h

lea si, FILE\_NAME

mov al, ' '

repe scasb

dec di

WRITE\_CYCLE:

mov ax, [di]

mov [si], ax

inc di

inc si

cmp [di], 0dh

jne WRITE\_CYCLE

mov [si], 0

lea dx, FILE\_NAME

mov ah, 9

int 21h

ret

GET\_FILE\_NAME endp

;IS TXT FILE;

IS\_TXT\_FILE proc

sub si, 4

lea di, TXT\_STRING

mov cx, 4

repe cmpsb

jne WRONG\_PARAMETR

ret

IS\_TXT\_FILE endp

;READ FROM FILE;

READ\_FROM\_FILE proc

mov ah, 3fh

mov bx, FILE\_DESKRIPTOR

xor cx, cx

mov cl, BUFFER\_SIZE

lea dx, BUFFER

int 21h

cmp al, 0

je EXIT

mov BYTES\_WAS\_READ, al

ret

READ\_FROM\_FILE endp

;OUTPUT BUFFER STRING;

OUTPUT\_BUFFER proc

xor cx, cx

mov cl, BYTES\_WAS\_READ

lea di, BUFFER

mov ax, 0b800h

mov es, ax

mov dh, 00000111b

mov bx, FILE\_POINTER

FOR1:

mov dl, [di]

mov es:[bx], dx

add bx, 2

inc di

push cx

mov cx, 0fffh

cmp bx, 4000

jl PAUSE

call CLEAR

mov bx, 160

PAUSE:

cmp CLOSE, 1

je EXIT

cmp STOP, 1

je PAUSE

loop pause

pop cx

cmp [di], 13

je REPLACE\_ON\_SPACE

cmp [di], 10

jne END\_OF\_LOOP

REPLACE\_ON\_SPACE:

mov [di], ' '

END\_OF\_LOOP:

loop FOR1

mov FILE\_POINTER, bx

ret

OUTPUT\_BUFFER endp

;CLEAR ALL SCREEN;

CLEAR\_ALL\_SCREEN proc

call CLEAR

xor bx, bx

mov cx, 80

clear\_cycle1:

mov es:[bx], dx

add bx, 2

loop clear\_cycle1

ret

CLEAR\_ALL\_SCREEN endp

;CLEAR;

CLEAR proc

mov ax, 0b800h

mov es, ax

mov bx, 160

mov cx, 2000

mov dl, ' '

clear\_cycle:

mov dh, 00000111b

mov es:[bx], dx

add bx, 2

loop clear\_cycle

ret

CLEAR endp

;OPEN FILE;

OPEN\_FILE proc

mov ah, 3dh

mov al, 0

lea dx, FILE\_NAME

int 21h

mov FILE\_DESKRIPTOR, ax

jc ERROR

ret

ERROR:

lea dx, OPEN\_FILE\_ERROR

mov ah, 9

int 21h

jmp EXIT

OPEN\_FILE endp

;CLOSE FILE;

CLOSE\_FILE proc

mov ah, 3Eh

mov bx, FILE\_DESKRIPTOR

int 21h

ret

CLOSE\_FILE endp

;NEW INT 9;

NEW\_INT9 proc far

push es

in al, 60h

cmp al, 0AEh

je EOI

cmp al, 9DH

je EOI

cmp al, 1

je SET\_EXIT\_FLAG

cmp al, 2Eh

jne RESET

CTRL\_IS\_PRESSED:

cmp PREW\_SCAN\_CODE, 1Dh

jne RESET

int 1bh

jmp EOI

SET\_EXIT\_FLAG:

mov cs:CLOSE, 1

jmp EOI

RESET:

mov cs:STOP, 0

EOI:

mov PREW\_SCAN\_CODE, al

mov al, 20h

out 20h, al

pop es

iret

NEW\_INT9 endp

;NEW INT 1B;

NEW\_INT1B proc

mov cs:STOP, 1

iret

NEW\_INT1b endp

;INSTALL NEW HANDLER;

INSTALL\_NEW\_HANDLER proc

mov ax, 3509h

int 21h

mov word ptr cs:OLD\_INT9 + 2, es

mov word ptr cs:OLD\_INT9, bx

mov ax, 351Bh

int 21h

mov word ptr cs:OLD\_INT1B + 2, es

mov word ptr cs:OLD\_INT1B, bx

push cs

pop ds

mov ax, 2509h

lea dx, NEW\_INT9

int 21h

mov ax, 251Bh

lea dx, NEW\_INT1B

int 21h

ret

INSTALL\_NEW\_HANDLER endp

;INSTALL OLD HANDLER;

INSTALL\_OLD\_HANDLER proc

mov dx, word ptr cs:OLD\_INT9

mov ds, word ptr cs:OLD\_INT9 + 2

mov ax, 2509h

int 21h

mov dx, word ptr cs:OLD\_INT1B

mov ds, word ptr cs:OLD\_INT1B + 2

mov ax, 251Bh

int 21h

ret

INSTALL\_OLD\_HANDLER endp

;DATA;

OLD\_INT9 dd ?

OLD\_INT1B dd ?

STOP db 0

CLOSE db 0

PREW\_SCAN\_CODE db 0

FILE\_POINTER dw 160

FILE\_DESKRIPTOR dw 0

COUNTER db 0

FILE\_NAME db 50 dup('$')

TXT\_STRING db ".txt", 0

BUFFER db 100 dup('$')

BUFFER\_SIZE db 100

BYTES\_WAS\_READ db 0

OPEN\_FILE\_ERROR db "ERROR WITH OPENING FILE$"

PARAMETR\_ERROR db "PARAMETR MUST BE NAME OF TXT FILE$"

MESSAGE\_TO\_EXIT db "PRESS ESC TO EXIT FROM PROGRAM$"

end start