

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация ЭВМ и систем»
Тема: Написание собственного прерывания

Студент гр. 1383

Кошкин Е.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Получение навыков создания собственных прерываний на языке Ассемблера.

Задание.

Вариант 12. (2е)

2 - 60h - прерывание пользователя - должно генерироваться в программе.

Е - Выполнить чтение и вывод на экран отсчета часов реального времени из памяти CMOS (в формате BCD).

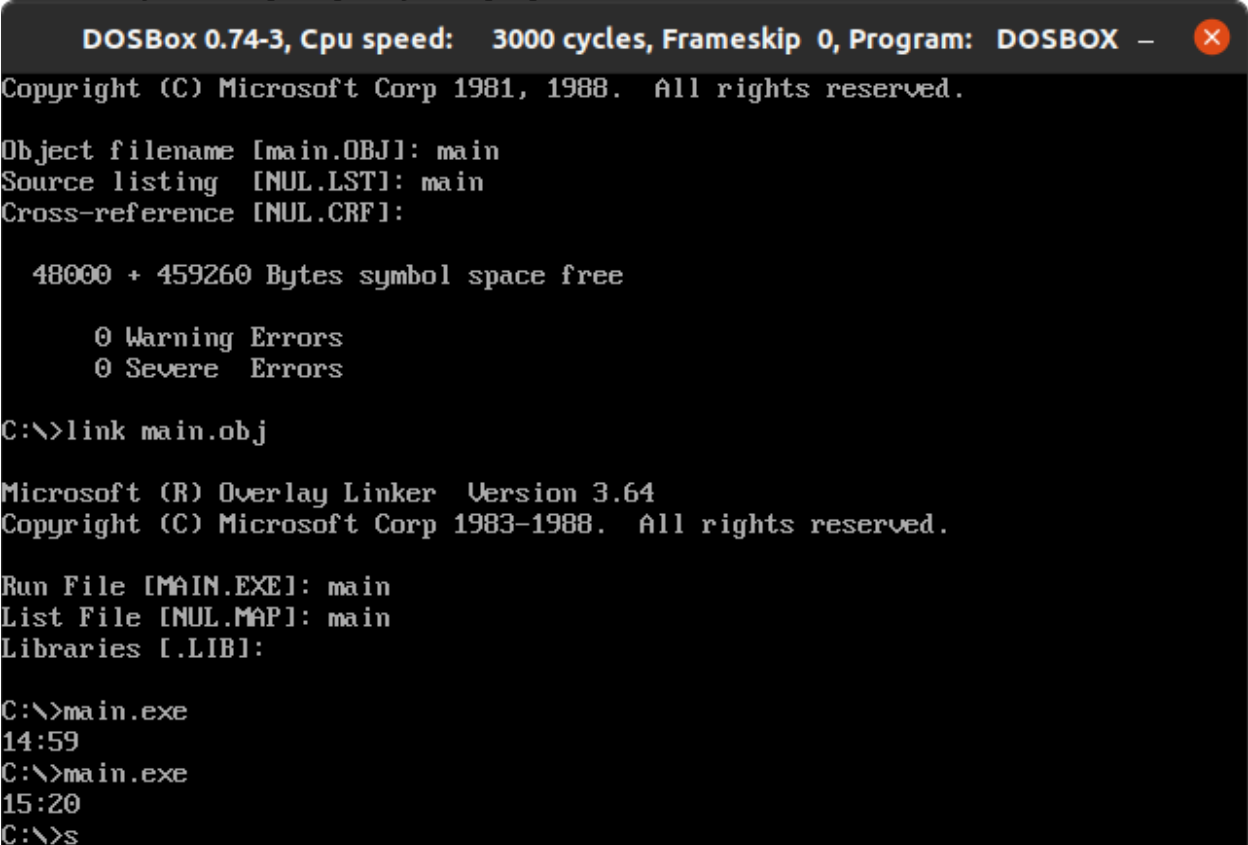
Выполнение работы.

Создаются переменные KEEP_CS и KEEP_IP для хранения адреса вектора старого прерывания для его последующего восстановления. В функции main с помощью функции 35h прерывания 21h в выше описанные переменные сохраняется адрес старого вектора. Далее в таблицу векторов записывается новый адрес собственного прерывания SUBR_INT с помощью функции 25h прерывания 21h.

После вызова нового прерывания восстанавливается адрес старого прерывания в таблице векторов с помощью функции 25h прерывания 21h.

Прерывание SUBR_INT считывает реальное время с помощью функции 02h прерывания 1Ah, которая записывает время в регистр cx в BCD-формате. Далее смещением и логическим И цифры времени преобразуются в их ASCII-коды. Функцией 02h прерывания 21h время выводится в формате ЧЧ:ММ.

Рисунок 1. Пример запуска программы



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX -
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [main.OBJ]: main
Source listing [NUL.LST]: main
Cross-reference [NUL.CRF]:

48000 + 459260 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link main.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [MAIN.EXE]: main
List File [NUL.MAP]: main
Libraries [.LIB]:

C:\>main.exe
14:59
C:\>main.exe
15:20
C:\>s
```

Выводы.

Получены навыки создания собственных прерываний на языке Ассемблера.

Разработана программа, считывающая реальное время из системы и выводящая его на экран.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

main.asm:

DATA SEGMENT

KEEP_CS dw 0

KEEP_IP dw 0

DATA ENDS

AStack SEGMENT STACK

db 1024 DUP (?)

AStack ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

SUBR_INT PROC FAR

push ax

push cx

push bx

push dx

; getting time

mov ah, 02h

int 1Ah

sub bx, bx

; printing time

mov bx, cx

shr bx, 1

shr bx, 1

shr bx, 1

shr bx, 1

and bx, 0F0Fh

and cx, 0F0Fh

add bx, 3030h

add cx, 3030h

mov dl, bh

int 21h

mov dl, ch

```

    int 21h
    mov dl, ':'
    int 21h
    mov dl, bl
    int 21h
    mov dl, cl
    int 21h

    pop dx
    pop bx
    pop cx
    pop ax
    mov al, 20h
    out 20h, al
    iret
SUBR_INT ENDP

```

```

Main PROC FAR
    ; initializing
    push ds
    sub ax, ax
    push ax
    mov ax, DATA
    mov ds, ax
    ; saving previous interrupt
    mov ah, 35h
    mov al, 60h
    int 21h
    mov KEEP_IP, bx
    mov KEEP_CS, es

    ; setting new interrupt address
    push ds
    mov dx, offset SUBR_INT
    mov ax, seg SUBR_INT
    mov ds, ax
    mov ah, 25h
    mov al, 60h
    int 21h
    pop ds

    ; calling new interrupt

```

```
int 60h

; restore previous interrupt
cli
push ds
mov dx, KEEP_IP
mov ax, KEEP_CS
mov ds, ax
mov ah, 25h
mov al, 60h
int 21h
pop ds
sti
ret
Main ENDP
CODE ENDS
END Main
```

main.lst:

0000 DATA SEGMENT

0000 0000 KEEP_CS dw 0

0002 0000 KEEP_IP dw 0

0004 DATA ENDS

0000 AStack SEGMENT STACK

0000 0400[db 1024 DUP (?)

??

]

0400 AStack ENDS

0000 CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

0000 SUBR_INT PROC FAR

0000 50 push ax

0001 51 push cx

0002 53 push bx

0003 52 push dx

; getting time

0004 B4 02 mov ah, 02h

0006 CD 1A int 1Ah

0008	2B DB	sub bx, bx
; printing time		
000A	8B D9	mov bx, cx
000C	D1 EB	shr bx, 1
000E	D1 EB	shr bx, 1
0010	D1 EB	shr bx, 1
0012	D1 EB	shr bx, 1
0014	81 E3 0F0F	and bx, 0F0Fh
0018	81 E1 0F0F	and cx, 0F0Fh
001C	81 C3 3030	add bx, 3030h
0020	81 C1 3030	add cx, 3030h
0024	8A D7	mov dl, bh
0026	CD 21	int 21h
0028	8A D5	mov dl, ch
002A	CD 21	int 21h
002C	B2 3A	mov dl, ':'
002E	CD 21	int 21h
0030	8A D3	mov dl, bl
0032	CD 21	int 21h
0034	8A D1	mov dl, cl
0036	CD 21	int 21h
0038	5A	pop dx
0039	5B	pop bx
003A	59	pop cx
003B	58	pop ax
003C	B0 20	mov al, 20h


```
003E E6 20                out 20h, al
0040 CF                  iret
0041                    SUBR_INT ENDP

0041                    Main PROC FAR
                        ; initializing
0041 1E                  push ds
0042 2B C0                sub ax, ax
0044 50                  push ax
0045 B8 ---- R           mov ax, DATA
0048 8E D8                mov ds, ax

                        ; saving previous interrupt
004A B4 35                mov ah, 35h
004C B0 60                mov al, 60h
004E CD 21                int 21h
0050 89 1E 0002 R        mov KEEP_IP, bx
0054 8C 06 0000 R        mov KEEP_CS, es

                        ; setting new interrupt address
0058 1E                  push ds
0059 BA 0000 R           mov dx, offset SUBR_INT
005C B8 ---- R           mov ax, seg SUBR_INT
005F 8E D8                mov ds, ax
0061 B4 25                mov ah, 25h
```

0063	B0 60	mov al, 60h
0065	CD 21	int 21h
0067	1F	pop ds
		; calling new interrupt
0068	CD 60	int 60h
		; restore previous interrupt
006A	FA	cli
006B	1E	push ds
006C	8B 16 0002 R	mov dx, KEEP_IP
0070	A1 0000 R	mov ax, KEEP_CS
0073	8E D8	mov ds, ax
0075	B4 25	mov ah, 25h
0077	B0 60	mov al, 60h
0079	CD 21	int 21h
007B	1F	pop ds
007C	FB	sti
007D	CB	ret
007E		Main ENDP
007E		CODE ENDS
		END Main

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0400	PARA		STACK
CODE	007E	PARA		NONE
DATA	0004	PARA		NONE

Symbols:

N a m e	Type	Value	Attr			
KEEP_CS	L WORD	0000	DATA			
KEEP_IP	L WORD	0002	DATA			
MAIN	F PROC	0041	CODE	Length	=	
003D						
SUBR_INT	F PROC	0000	CODE	Length	=	
0041						
@CPU	TEXT	0101h				
@FILENAME	TEXT	main				
@VERSION	TEXT	510				

98 Source Lines

98 Total Lines

12 Symbols

48000 + 459260 Bytes symbol space free

0 Warning Errors

0 Severe Errors