

**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.**

Студентка гр. 1303

Хулап О.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Написать собственное прерывание в соответствии с вариантом на языке Ассемблер.

Задание.

Вариант 26: 4Е

4 - 16h - прерывание от клавиатуры (по заданному скан-коду клавиши делать действия А-Г, указанные ниже).

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

Выполнение работы

В Main загружается в DS адрес начала сегмента данных, сохраняется сегмент и смещение текущего прерывания с помощью функции 35 прерывания 21h. Функция 25h прерывания 21h устанавливает вектор прерывания 16h, после чего вызывается созданное прерывание. После всех процедур необходимо восстановить старый вектор прерывания.

SUBR_INT – процедура прерывания. Все изменяемые переменные сохраняются в стек. 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.

Тестирование

```
C:\>MASM.EXE
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Source filename [.ASM]: lab5.asm
Line invalid, start again
Source filename [.ASM]: lab5.asm
Object filename [lab5.OBJ]:
Source listing [NUL.LST]: lab5.lst
Cross-reference [NUL.CRF]:

48018 + 461289 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>_
```

```
C:\>lab5.exe  
15:32  
C:\>_
```

Выводы

В ходе выполнения лабораторной работы были получены навыки написания программы с собственным прерыванием на языке Ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.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
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX

    MOV AH, 35H
    MOV AL, 16H
    INT 21H
    MOV KEEP_IP, BX
    MOV KEEP_CS, ES

    try:
        mov ah, 0
        int 16h
        cmp ah, 12h
        jnz try

    PUSH DS
    MOV DX, OFFSET SUBR_INT
    MOV AX, SEG SUBR_INT
    MOV DS, AX
    MOV AH, 25H
    MOV AL, 16H
    INT 21H
    POP DS

    int 16H

    CLI
    PUSH DS
    MOV DX, KEEP_IP
    MOV AX, KEEP_CS
    MOV DS, AX
    MOV AH, 25H
    MOV AL, 16H
    INT 21H
    POP DS
    STI

    MOV AH, 4Ch
    INT 21h
Main ENDP

```

```
CODE ENDS  
      END Main
```

ПРИЛОЖЕНИЕ Б
ЛИСТИНГ ФАЙЛ ИСХОДНОЙ ПРОГРАММЫ

lab5.lst:

#Microsoft (R) Macro Assembler Version 5.10
11/29/22 15:17:4

Page

1-1

```
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

1-2

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

0041                               Main PROC  FAR
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

004A  B4 35                      MOV AH, 35H
004C  B0 16                      MOV AL, 16H
004E  CD 21                      INT 21H
0050  89 1E 0002 R              MOV KEEP_IP, BX
0054  8C 06 0000 R              MOV KEEP_CS, ES

0058                               try:
0058  B4 00                      mov ah, 0
005A  CD 16                      int 16h
005C  80 FC 12                      cmp ah, 12h
005F  75 F7                      jnz try

0061  1E                          PUSH DS
0062  BA 0000 R                  MOV DX, OFFSET SUBR_INT
```

0065	B8 ---- R	MOV AX, SEG SUBR_INT
0068	8E D8	MOV DS, AX
006A	B4 25	MOV AH, 25H
006C	B0 16	MOV AL, 16H
006E	CD 21	INT 21H
0070	1F	POP DS
0071	CD 16	int 16H
0073	FA	CLI
0074	1E	PUSH DS
0075	8B 16 0002 R	MOV DX, KEEP_IP
0079	A1 0000 R	MOV AX, KEEP_CS
007C	8E D8	MOV DS, AX
007E	B4 25	MOV AH, 25H
0080	B0 16	MOV AL, 16H
0082	CD 21	INT 21H
0084	1F	POP DS
0085	FB	STI
0086	B4 4C	MOV AH, 4Ch
0088	CD 21	INT 21h
008A		Main ENDP
008A		CODE ENDS
		END Main

#Microsoft (R) Macro Assembler Version 5.10

11/29/22 15:17:4

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	0400	PARA	STACK
CODE	008A	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 = 0049
SUBR_INT	F PROC	0000	CODE Length = 0041
TRY	L NEAR	0058	CODE
@CPU	TEXT	0101h	
@FILENAME	TEXT	1ab5	
@VERSION	TEXT	510	

104 Source Lines

104 Total Lines

13 Symbols

48018 + 461289 Bytes symbol space free

0 Warning Errors

0 Severe Errors