

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: Разработка собственного прерывания.

Студентка гр. 0383

Преподаватель

Ханина М.И.

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Вариант 3d:

Создать собственное прерывание с номером 23h - прерывание, генерируемое при нажатии клавиш Control + C и выполняющее чтение и вывод на экран отсчета системных часов.

Выполнение работы.

Для хранения сегмента и смещения прерывания создаются переменные KEEP_CS и KEEP_IP. В сегменте кода создан обработчик прерывания my_int, процедуры для вывода часа, минут и секунд, а для вывода двоеточия между ними создана процедура colon. Также задается собственный стек для обработчика прерывания. Процедура two_digit помогает вывести текущее время в виде двузначного числа на экран.

Для вывода времени используется функция MS-DOS которая возвращает системное время. После вызова 21h прерывания в регистры возвращается: ch - час; cl - минуты; dh - секунды.

Прерывание меняется следующим образом: с помощью функции 35h прерывания 21h получаем необходимый вектор прерывания 23h. Затем записываем этот вектор в переменные KEEP_CS и KEEP_IP. Далее с помощью функции 25h прерывания 21h восстанавливаем предыдущие значения.

Результаты тестирования представлены в табл. 1.

Тексты исходного файла программы см. в приложении А.

Тексты диагностических сообщений см. в приложении Б.

Таблица 1. Проверка работы программы.

№	Входные данные	Выходные данные	Комментарий
1	Нажатие Control + C в 16:55:10	16:55:10	Программа работает корректно
2	Нажатие другой комбинации клавиш	Ничего не происходит	Программа работает корректно

Выводы.

В ходе выполнения данной лабораторной работы была изучена работа с прерываниями на языке Ассемблер.

ПРИЛОЖЕНИЕ А
ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММЫ

Название файла: **lb5.asm**

AStack SEGMENT STACK

DB 1024 DUP(?)

AStack ENDS

DATA SEGMENT

KEEP_CS DW 0 ;хранение сегмента

KEEP_IP DW 0 ;хранение смещения вектора прерывания

DATA ENDS

CODE SEGMENT

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

my_int PROC FAR

jmp start

KEEP_SS DW 0

KEEP_SP DW 0

KEEP_AX DW 0

IStack DW 30 DUP(?)

hours proc

mov al,ch

call two_digit

ret

hours endp

minutes proc

```
    mov al,cl
    call two_digit
    ret
```

```
minutes endp
```

```
seconds proc
```

```
    mov al,dh
    call two_digit
    ret
```

```
seconds endp
```

```
colon proc
```

```
    mov ah,2
    mov dl,':'
    int 21h
    ret
```

```
colon endp
```

```
two_digit proc ;Процедура вывода двузначного числа.
```

```
    push dx
    aam
    add ax,3030h
    mov dl,ah
    mov dh,al
    mov ah,02
    int 21h
    mov dl,dh
    int 21h
    pop dx
```

```
        ret
two_digit endp
```

start:

```
    MOV KEEP_SP, SP
    MOV KEEP_AX, AX
    MOV AX, SS
    MOV KEEP_SS, AX
    MOV AX, KEEP_AX
    MOV SP, OFFSET start
    MOV SS, AX
```

```
    push AX      ; сохранение изменяемых регистров
    push DX
```

```
;-----
```

```
    push ax
```

```
        mov ah, 2ch ; помещаем код функции 2ch - получить сис время
```

```
        int 21h ; вызов прерывания ms-dos
```

```
        pop ax
```

```
    call hours
```

```
    call colon
```

```
    call minutes
```

```
    call colon
```

```
    call seconds
```

```
;-----
```

```
    pop DX
```

```
    pop AX
```

```
    MOV KEEP_AX, AX
```

```
MOV SP, KEEP_SP
MOV AX, KEEP_SS
MOV SS, AX
MOV AX, KEEP_AX
```

```
mov al,20h
out 20h,al
iret
```

```
my_int ENDP
```

```
MAIN PROC FAR
```

```
MOV AX, DATA
MOV DS, AX ; сохраняем вектор прерывания
MOV AH, 35H ; функция получения вектора
MOV AL, 23H ; номер нужного вектора
INT 21H
MOV KEEP_IP, BX ; запоминание смещения
MOV KEEP_CS, ES ; и сегмента вектора прерывания
```

```
PUSH DS
MOV DX, OFFSET my_int ; смещение для процедуры в DX
MOV AX, SEG my_int ; сегмент процедуры
MOV DS, AX; помещаем в DS
MOV AH, 25H; функция установки вектора
MOV AL, 23H; номер вектора
INT 21H; меняем прерывание
POP DS
```

```
ctrl_c:
```

```
mov ah, 0
int 16h
cmp al, 3 ;код символа после нажатия
jne ctrl_c
```

```
INT 23H
```

```
CLI
```

```
PUSH DS
```

```
MOV DX, KEEP_IP
```

```
MOV AX, KEEP_CS
```

```
MOV DS, AX
```

```
MOV AH, 25H
```

```
MOV AL, 23H
```

```
INT 21H
```

```
; восстанавливаем старый вектор прерывания
```

```
POP DS
```

```
STI
```

```
MOV AH, 4CH
```

```
INT 21H
```

```
MAIN ENDP
```

```
CODE ENDS
```

```
END MAIN
```


ПРИЛОЖЕНИЕ Б
ТЕКСТЫ ДИАГНОСТИЧЕСКИХ ФАЙЛОВ ПРОГРАММЫ

Название файла: **LB5.LST**

Microsoft (R) Macro Assembler Version 5.10

11/29/21 12:18:5

Page 1-1

```
0000          AStack SEGMENT STACK
0000 0400[          DB 1024 DUP(?)
      ??
      ]

0400          AStack ENDS

0000          DATA      SEGMENT
0000 0000          KEEP_CS DW 0 ;хранение сегмента
0002 0000          KEEP_IP DW 0 ;хранение смещения вектора прерывания
0004          DATA      ENDS

0000          CODE      SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

0000          my_int PROC FAR
0000 EB 70 90          jmp start
0003 0000          KEEP_SS DW 0
0005 0000          KEEP_SP DW 0
0007 0000          KEEP_AX DW 0
```

```
0009 001E[                IStack DW 30 DUP(?)
      ???
      ]
```

```
0045                hours proc
0045 8A C5            mov al,ch
0047 E8 005E R        call two_digit
004A C3              ret
004B                hours endp
```

```
004B                minutes proc
004B 8A C1            mov al,cl
004D E8 005E R        call two_digit
0050 C3              ret
0051                minutes endp
```

```
0051                seconds proc
0051 8A C6            mov al,dh
0053 E8 005E R        call two_digit
0056 C3              ret
0057                seconds endp
```

```
0057                colon proc
0057 B4 02            mov ah,2
0059 B2 3A            mov dl,':'
005B CD 21            int 21h
005D C3              ret
005E                colon endp
```

005E two_digit proc ;Процедура вывода
 ° двузначного числа.

Microsoft (R) Macro Assembler Version 5.10

11/29/21 12:18:5

Page 1-2

```
005E 52                      push dx
005F D4 0A                      aam
0061 05 3030                      add ax,3030h
0064 8A D4                      mov dl,ah
0066 8A F0                      mov dh,al
0068 B4 02                      mov ah,02
006A CD 21                      int 21h
006C 8A D6                      mov dl,dh
006E CD 21                      int 21h
0070 5A                      pop dx
0071 C3                      ret
0072                      two_digit endp

0072                      start:
0072 2E: 89 26 0005 R                      MOV KEEP_SP, SP
0077 2E: A3 0007 R                      MOV KEEP_AX, AX
007B 8C D0                      MOV AX, SS
007D 2E: A3 0003 R                      MOV KEEP_SS, AX
0081 2E: A1 0007 R                      MOV AX, KEEP_AX
0085 BC 0072 R                      MOV SP, OFFSET start
0088 8E D0                      MOV SS, AX
```

```

008A 50          push AX    ; сохранение измеИ
                          1/2яемых регистров
008B 52          push DX
                          ;-----
008C 50          push ax
008D B4 2C          mov ah, 2ch  ; помещаем код фэ
                          фнкции 2ch - получить сис врИ
                          ммя
008F CD 21          int 21h ; вызов прерывания m
                          s-dos
0091 58          pop ax
0092 E8 0045 R      call hours
0095 E8 0057 R      call colon
0098 E8 004B R      call minutes
009B E8 0057 R      call colon
009E E8 0051 R      call seconds
                          ;-----

00A1 5A          pop DX
00A2 58          pop AX
00A3 2E: A3 0007 R   MOV KEEP_AX, AX
00A7 2E: 8B 26 0005 R   MOV SP, KEEP_SP
00AC 2E: A1 0003 R   MOV AX, KEEP_SS
00B0 8E D0          MOV SS, AX
00B2 2E: A1 0007 R   MOV AX, KEEP_AX

00B6 B0 20          mov al,20h
00B8 E6 20          out 20h,al
00BA CF            iret

```

00BB my_int ENDP

00BB MAIN PROC FAR

Microsoft (R) Macro Assembler Version 5.10

11/29/21 12:18:5

Page 1-3

00BB B8 ---- R MOV AX, DATA

00BE 8E D8 MOV DS, AX ; сохраняем вектор прерывания

00C0 B4 35 MOV AH, 35H ; функция получения вектора

00C2 B0 23 MOV AL, 23H ; номер нужного вектора

00C4 CD 21 INT 21H

00C6 89 1E 0002 R MOV KEEP_IP, BX ; запоминание смещения

00CA 8C 06 0000 R MOV KEEP_CS, ES ; и сегмента вектора прерывания

00CE 1E PUSH DS

00CF BA 0000 R MOV DX, OFFSET my_int ; смещение для процедуры в DX

00D2 B8 ---- R MOV AX, SEG my_int ; сегмент процедуры

00D5 8E D8 MOV DS, AX; помещаем в DS

00D7 B4 25 MOV AH, 25H; функция установки вектора

00D9 B0 23 MOV AL, 23H; номер вектора

00DB	CD 21	INT 21H; меняем прерывание
00DD	1F	POP DS
00DE		ctrl_c:
00DE	B4 00	mov ah, 0
00E0	CD 16	int 16h
00E2	3C 03	cmp al, 3 ;код символа послИ
		μ нажатия
00E4	75 F8	jne ctrl_c
00E6	CD 23	INT 23H
00E8	FA	CLI
00E9	1E	PUSH DS
00EA	8B 16 0002 R	MOV DX, KEEP_IP
00EE	A1 0000 R	MOV AX, KEEP_CS
00F1	8E D8	MOV DS, AX
00F3	B4 25	MOV AH, 25H
00F5	B0 23	MOV AL, 23H
00F7	CD 21	INT 21H
		; восстанавливаем стары
		й вектор прерывания
00F9	1F	POP DS
00FA	FB	STI
00FB	B4 4C	MOV AH, 4CH
00FD	CD 21	INT 21H
00FF		MAIN ENDP

00FF

CODE ENDS

END MAIN

Microsoft (R) Macro Assembler Version 5.10

11/29/21 12:18:5

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0400	PARA		STACK
CODE	00FF	PARA		NONE
DATA	0004	PARA		NONE

Symbols:

N a m e	Type	Value	Attr
COLON	N PROC	0057	CODE Length = 0007
CTRL_C	L NEAR	00DE	CODE
HOURS	N PROC	0045	CODE Length = 0006
ISTACK	L WORD	0009	CODE Length = 001E
KEEP_AX	L WORD	0007	CODE
KEEP_CS	L WORD	0000	DATA
KEEP_IP	L WORD	0002	DATA
KEEP_SP	L WORD	0005	CODE

KEEP_SS L WORD 0003 CODE

MAIN F PROC 00BB CODE Length = 0044

MINUTES N PROC 004B CODE Length = 0006

MY_INT F PROC 0000 CODE Length = 00BB

SECONDS N PROC 0051 CODE Length = 0006

START L NEAR 0072 CODE

TWO_DIGIT N PROC 005E CODE Length = 0014

@CPU TEXT 0101h

@FILENAME TEXT 1b5

@VERSION TEXT 510

138 Source Lines

138 Total Lines

23 Symbols

48034 + 459226 Bytes symbol space free

0 Warning Errors

0 Severe Errors