

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

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

Студент гр. 9383

Арутюнян С.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Содержание

1. Цель работы.....	3
2. Задание. Вариант 1А.....	3
3. Текст листинга lab5.lst.....	4
4. Текст программы lab5.asm.....	8
Выводы.....	10

1. Цель работы

Научиться работать с прерываниями и создавать собственные.

2. Задание. Вариант 1А.

Заменить обработчик прерывания с номером 1Ch (прерывание от часов, генерируется ОС 18 раз в секунду) на обработчик прерывания, выводящий на экран приветственное сообщение.

3. Текст листинга lab5.lst

#Microsoft (R) Macro Assembler Version 5.10

11/25/20 21:20:1

Page 1-1

```
0000          AStack SEGMENT STACK
0000 0100[          DW 256 DUP(?)      ]

0200          AStack ENDS
0000          DATA SEGMENT
0000 0000          INT_IP DW 0
0002 0000          INT_CS DW 0
0004 48 65 6C 6C 6F 2C GREETINGS db "Hello, World!$"
      20 57 6F 72 6C 64
      21 24
0012          DATA ENDS
0000          CODE SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack
0000          my_interrupt PROC FAR
0000 50          push ax
0001 52          push dx
0002 B4 09          mov ah, 09h
0004 BA 0004 R      mov dx, offset GREETINGS
0007 CD 21          int 21h
0009 5A          pop dx
000A 58          pop ax
```

```

000B B0 20          mov al, 20h
000D E6 20          out 20h, al
000F CF            iret
0010              my_interrupt ENDP
0010              Main PROC FAR
0010 B8 ---- R      mov ax, DATA
0013 8E D8          mov ds, ax
0015 B4 35          mov ah, 35h
0017 B0 1C          mov al, 1ch
0019 CD 21          int 21h
001B 89 1E 0000 R   mov INT_IP, bx
001F 8C 06 0002 R   mov INT_CS, es
0023 1E            push ds
0024 BA 0000 R      mov dx, offset my_interrupt

```

#Microsoft (R) Macro Assembler Version 5.10

11/25/20 21:20:1

Page 1-2

```

0027 B8 ---- R      mov ax, seg my_interrupt
002A 8E D8          mov ds, ax
002C B4 25          mov ah, 25h
002E B0 1C          mov al, 1ch
0030 CD 21          int 21h
0032 1F            pop ds
0033 CD 1C          int 1ch
0035 FA            cli
0036 1E            push ds

```

```

0037 8B 16 0000 R      mov dx, INT_IP
003B A1 0002 R      mov ax, INT_CS
003E 8E D8          mov ds, ax
0040 B4 25          mov ah, 25h
0042 B0 1C          mov al, 1ch
0044 CD 21          int 21h
0046 1F            pop ds
0047 FB            sti
0048 B4 4C          mov ah, 4ch
004A CD 21          int 21h
004C              Main ENDP
004C              CODE ENDS

```

END Main

#Microsoft (R) Macro Assembler Version 5.10

11/25/20 21:20:1

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0200	PARA	STACK
CODE	004C	PARA	NONE
DATA	0012	PARA	NONE

Symbols:

N a m e	Type	Value	Attr
GREETINGS	L BYTE	0004	DATA
INT_CS	L WORD	0002	DATA
INT_IP	L WORD	0000	DATA

MAIN F PROC 0010 CODE Length = 003C
MY_INTERRUPT F PROC 0000 CODE Length =
0010
@CPU TEXT 0101h
@FILENAME TEXT lab5
@VERSION TEXT 510
76 Source Lines
76 Total Lines
13 Symbols
48000 + 459260 Bytes symbol space free
0 Warning Errors
0 Severe Errors

4. Текст программы lab5.asm

```
AStack SEGMENT STACK
```

```
    dw 256 DUP(?)    ; 1 килобайт
```

```
AStack ENDS
```

```
DATA SEGMENT
```

```
INT_IP dw 0          ; IP прерывания
```

```
INT_CS dw 0          ; CS прерывания
```

```
GREETINGS db "This interrupt is working!", 10, 13, '$'
```

```
DATA ENDS
```

```
CODE SEGMENT
```

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

```
my_interrupt PROC FAR
```

```
    jmp int_start
```

```
    SAVED_SP dw 0
```

```
    SAVED_SS dw 0
```

```
    SAVED_AX dw 0
```

```
    INT_STACK dw 16 DUP(?)
```

```
int_start:
```

```
    mov SAVED_SP, sp
```

```
    mov SAVED_AX, ax
```

```
    mov ax, ss
```

```
    mov SAVED_SS, ax
```

```
    mov ax, SAVED_AX
```

```
    mov sp, offset int_start
```

```
    mov ax, seg INT_STACK
```

```
    mov ss, ax
```



```
push ax
push dx
```

```
mov ah, 09h
mov dx, offset GREETINGS
int 21h
```

```
pop dx
pop ax
```

```
mov SAVED_AX, ax
mov sp, SAVED_SP
mov ax, SAVED_SS
mov ss, ax
mov ax, SAVED_AX
```

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

```
my_interrupt ENDP
```

```
Main PROC FAR
```

```
mov ax, DATA
mov ds, ax
```

```
mov ah, 35h
mov al, 1ch
int 21h
```

```
mov INT_IP, bx ; сохраняем информацию о прерывании
mov INT_CS, es
```

```

push ds
mov dx, offset my_interrupt ; суем в dx смещение процедуры в dx
mov ax, seg my_interrupt   ; суем в ax сегмент процедуры
mov ds, ax                 ; помещаем сегмент в ds
mov ah, 25h                ; функция установки вектора прерывания
mov al, 1ch                ; номер вектора
int 21h                    ; запускаем изменение прерывания
pop ds

```

; цикл, пока не нажмем q

input:

```

    mov ah, 1
    int 21h
    cmp al, 71h
    jne input

```

cli

```

push ds
mov dx, INT_IP
mov ax, INT_CS
mov ds, ax
mov ah, 25h
mov al, 1ch      ; номер прерывания
int 21h          ; восстанавливаем старый вектор
pop ds
sti

```

```

mov ah, 4ch
int 21h

```

Main ENDP

CODE ENDS

END Main

Выводы

В процессе выполнения лабораторной работы было изучено представление и обработка символьной информации с использованием строковых команд.