

**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: Создание собственных прерываний.**

Студент гр. 0383

Пенкин М.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### Вариант 8:

## Выполнение работы.

### Таблица 1. Проверка работы программы

[illegible]

	Hello world! Hello world! Hello world! End!
--	--

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении Б.

### **Выводы.**

В ходе выполнения данной лабораторной работы была изучена разработка собственных прерываний на языке Ассемблер.

## ПРИЛОЖЕНИЕ А

### ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: lr5.asm

```
DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения вектора прерывания

    HELLO    DB 'Hello World!',10,13,'$'
    MESEND   DB 'End!',10,13,'$'

DATA ENDS

AStack      SEGMENT  STACK
            DW 12 DUP(?) ; ♦:♦♦♦♦♦♦ 12 ♦ ♦♦♦♦♦♦
AStack      ENDS

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

SUBR_INT PROC FAR
    jmp start_proc
    KEEP_SS DW 0
    KEEP_SP DW 0
    KEEP_AX DW 0

    ;MESEND   DB 'End!',10,13,'$'
    BStack DW 12 DUP(?)
start_proc:

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

    MOV AX, KEEP_AX

    MOV SP, OFFSET start_proc

    MOV AX, seg BStack
    MOV SS, AX

    PUSH AX ; сохранение изменяемых регистров
    PUSH DX;

    MOV     DX, OFFSET HELLO
    MOV     AH,9
metka:
    int     21h ; Вызов функции DOS по прерыванию
loop metka ; Вывод сообщения заданное число раз
```

```

;mov al, 0
;mov ah, 86h
;mov cx, 0098h
;mov dx, 9680h
int 15h; Фиксированная задержка

```

```

                                MOV      DX, OFFSET MESEND ; Вывод сообщения о
завершении обработчика
                                MOV      AH, 9
int 21h

```

```

POP DX;
POP AX ; восстановление регистров

```

```

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

```

```

SUBR_INT ENDP

```

```

Main      PROC  FAR

```

```

                                push  DS          ;\  Сохранение адреса начала PSP в
стеке

```

```

                                sub    AX,AX      ; > для последующего восстановления по
                                push   AX          ;/  команде ret, завершающей
процедуру.

```

```

mov    AX,DATA                  ; Загрузка сегментного
mov    DS,AX                    ; регистра данных.

```

```

MOV AH, 35H ; функция получения вектора
MOV AL, 60H ; номер вектора
INT 21H ; возвращает текущее значение вектора
прерывания

```

```

MOV KEEP_IP, BX ; запоминание смещения
MOV KEEP_CS, ES ; и сегмента вектора прерывания

```

```

PUSH DS
MOV DX, OFFSET SUBR_INT ; смещение для процедуры в DX
MOV AX, SEG SUBR_INT ; сегмент процедуры
MOV DS, AX ; помещаем в DS
MOV AH, 25H ; функция установки вектора
MOV AL, 60H ; номер вектора
INT 21H ; меняем прерывание
POP DS

```

```

mov cx, 10
;mov dx, 3
    int 60H; вызов измененного прерывания

    CLI
    PUSH DS
    MOV DX, KEEP_IP
    MOV AX, KEEP_CS
    MOV DS, AX
    MOV AH, 25H
    MOV AL, 60H
    INT 21H ; восстанавливаем старый вектор прерывания
    POP DS
    STI

    RET
Main      ENDP
CODE      ENDS
          END Main

```

## ПРИЛОЖЕНИЕ Б

### ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr5.lst

```
DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения вектора прерывания
    TMP1 DW 0
    TMP2 DW 0
    TMP3 DW 0
    HELLO DB 'Hello World!',10,13,'$'
    MESEND DB 'End!',10,13,'$'
```

DATA ENDS

```
AStack SEGMENT STACK
    DW 12 DUP(?) ; ♦:♦♦♦♦♦♦ 12 ♦ ♦♦♦♦♦♦
AStack ENDS
```

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

```
SUBR_INT PROC FAR
    jmp start_proc

    ST_SS DW 0000
    ST_AX DW 0000
    ST_SP DW 0000
    IStack DW 30 DUP(?)
start_proc:
```

```
    mov ST_SP, SP
mov ST_AX, AX
mov AX, SS
mov ST_SS, AX
mov AX, IStack
mov SS, AX
mov AX, ST_AX
push ax
push ds
```

```
MOV DX, OFFSET HELLO
```

```

MOV     AH,9
metka:
int     21h ; Вызов функции DOS по прерыванию
loop metka ; Вывод сообщения заданное число раз

```

```

mov di,32
mov ah,0
int 1Ah
mov bx,dx; счетчик с момента сброса
Delay:
mov ah,0
int 1Ah
sub dx,bx
cmp di,dx
ja Delay;переход,если больше

```

```

MOV DX, OFFSET MESEND ;Выводсообщенияозавершении обработчика
MOV AH,9
int 21h
pop dx

```

```

        pop ax
mov ST_AX,AX
mov AX,ST_SS
mov SS,AX
mov SP,ST_SP
mov AX,ST_AX
mov al,20h
out 20h,al

```

```

iret
SUBR_INT ENDP

```

```

Main      PROC   FAR

```

```

        push    DS           ;\ Сохранение адреса начала PSP в
стеке

```

```

        sub     AX,AX        ; > для последующего восстановления по
        push    AX           ;/ команде ret, завершающей
процедуру.

```

```

        mov     AX,DATA      ; Загрузка сегментного
        mov     DS,AX        ; регистра данных.

```

```

        MOV AH, 35H ; функция получения вектора
        MOV AL, 60H ; номер вектора
        INT 21H ; возвращает текущее значение вектора
прерывания

```

```

        MOV KEEP_IP, BX ; запоминание смещения
        MOV KEEP_CS, ES ; и сегмента вектора прерывания

```

```

        PUSH DS
        MOV DX, OFFSET SUBR_INT ; смещение для процедуры в DX
        MOV AX, SEG SUBR_INT ; сегмент процедуры
        MOV DS, AX ; помещаем в DS
        MOV AH, 25H ; функция установки вектора
        MOV AL, 60H ; номер вектора

```



```

        INT 21H ; меняем прерывание
        POP DS

mov cx, 10
        int 60H; вызов измененного прерывания

        CLI
        PUSH DS
        MOV DX, KEEP_IP
        MOV AX, KEEP_CS
        MOV DS, AX
        MOV AH, 25H
        MOV AL, 60H
        INT 21H ; восстанавливаем старый вектор прерывания
        POP DS
        STI

        RET
Main      ENDP
CODE      ENDS
          END Main

```