

**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.**  
**ВАРИАНТ 22**

Студент гр. 1383

Харитонов Н.М

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить прерывания на Ассемблере и реализовать собственное прерывание.

### **Задание.**

4 – 16h - прерывание от клавиатуры (по заданному скан-коду клавиши делать действия A-G, указанные ниже)

A – Выполнить вывод сообщения на экран заданное число раз, после чего вставить фиксированную задержку и вывести сообщение о завершении обработчика.

### **Выполнение работы.**

В данной программе, в качестве сообщения для вывода, есть строка, хранящаяся в сегменте данных “Hello World!” с переносом на новую строку и символом окончания строки ‘\$’. После выполнения основной части программы, перед её завершением выводится сообщение “Completion of processing”.

В программе вызывается прерывание (16h) с клавиатуры, которое считывает скан-код клавиши до того момента, пока скан-код не будет соответствовать клавише “escape”.

Также в программе есть собственное (60h) прерывание, которое срабатывает после правильно нажатой клавиши. В нем происходит три вывода на экран сообщения с помощью цикла, затем, срабатывает секундная задержка и выводится на экран второе сообщение. Перед прерыванием с клавиатуры сохраняется старый вектор прерывания за счет функции 35 прерывания 21h. После всех выводов на экран и окончания работы собственного прерывания, восстанавливается старый вектор прерывания.

### **Выводы.**

В ходе выполнения работы были изучены прерывания, в том числе и прерывание 16h – прерывание от клавиатуры.



## ПРИЛОЖЕНИЕ А

### КОД ПРОГРАММЫ

#### **lr5.asm:**

```
AStack SEGMENT STACK
```

```
    db 1024 DUP(?)
```

```
AStack ENDS
```

```
DATA SEGMENT
```

```
    KEEP_CS dw 0
```

```
    KEEP_IP dw 0
```

```
    crlf db 0dh, 0ah, '$'
```

```
    mes1 db 'Hello World!',0dh, 0ah, '$'
```

```
    mes2 db 'Completion of processing$'
```

```
DATA ENDS
```

```
CODE SEGMENT
```

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

```
SUBR_INT PROC FAR
```

```
    PUSH AX
```

```
    push DX
```

```
    push bx
```

```
    push cx
```

```
    mov dx, offset mes1
```

```
    mov cx, 3
```

```
print_msg1:
```

```
    mov ah, 9h
```

```
    int 21h
```

```
    loop print_msg1
```

```

MOV      CX, 0FH
MOV      DX, 4240H
MOV      AH, 86H
INT      15H

mov dx, offset mes2
mov ah, 9h
int 21h

pop cx
pop bx
pop dx
POP AX
MOV AL, 20H
OUT 20H,AL
IRET
SUBR_INT ENDP

```

```

Main PROC FAR
    push ds
    xor ax, ax
    push ax
    mov ax, DATA
    mov ds, ax

    mov ah, 35h
    mov al, 60h
    int 21h

```

```

mov KEEP_IP, bx
mov KEEP_CS, es

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

loop1:
    mov ah, 00h
    int 16h
    cmp ah, 01h
    jne loop1
    mov ah, 25h
    int 60h

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