

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

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

Студент гр. 9383

\_\_\_\_\_

Корсунов А.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Изучить прерывания на языке ассемблере. Написать программу на основе изученного.

### **Текст задания.**

Вариант 2А

Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

1 - 1Ch - прерывание от часов - генерируется автоматически операционной системой 18 раз в сек;

2 - 60h - прерывание пользователя - должно генерироваться в программе;

3 - 23h - прерывание, генерируемое при нажатии клавиш Control+C ;

4 - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

Буква определяет действия, реализуемые программой обработки прерываний:

A - Печать сообщения на экране;

B - Выдача звукового сигнала;

C - Приостановить вывод на экран (вставить цикл задержки).

Замечание: для исключения возможного взаимного влияния системных и пользовательских прерываний рекомендуется отвести в программе под стек не менее 1K байт.

### **Ход работы:**

Для реализации поставленной задача была написана процедура SUBR\_INT, выводящая сообщение 'Hello world!', которое хранится в сегменте данных, на экран.

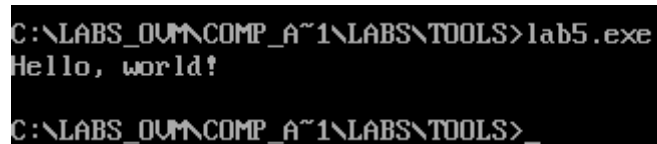
Программа, использующая новые программы обработки прерываний при своем завершении должна восстанавливать оригинальные векторы прерываний.

Функция 35 прерывания 21H возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX.

Для установки адреса нового обработчика прерывания в поле векторов прерываний используется функция 25H прерывания 21H, которая помещает заданные адреса сегмента и смещения обработчика в вектор прерывания с заданным номером.

Команды CLI и STI служат для установки или сброса флага прерываний, что позволяет включать или отключать реакцию на внешние прерывания. Команда CLI (Clear Interrupt flag) сбрасывает флаг IF в значение 0, что запрещает прерывания. Команда STI (Set Interrupt flag) устанавливает флаг IF в значение 1, что разрешает прерывания.

### **Тестирование:**



```
C:\LABS_OUM\COMP_A~1\LABS\TOOLS>lab5.exe
Hello, world!
C:\LABS_OUM\COMP_A~1\LABS\TOOLS>_
```

Рисунок 1 - Пример работы программы

### **Вывод:**

Изучены прерывания на языке ассемблере. Написана программа на основе изученного.

## Приложение А

AStack SEGMENT STACK

db 1024 DUP(?)

AStack ENDS

DATA SEGMENT

KEEP\_CS DW 0 ; для хранения сегмента

KEEP\_IP DW 0 ; и смещения вектора прерывания

message DB 'Hello, world!', 13, 10, '\$'

DATA ENDS

CODE SEGMENT

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

SUBR\_INT PROC FAR

ax1 dw 0

dx1 dw 0

mov ax1, ax

mov dx1, dx

MOV AH, 9h

MOV dx1, OFFSET message

INT 21h

MOV AL, 20h ; разрешение обработки прерываний

OUT 20h, AL ; более низкого уровня

mov ax, ax1

mov dx, dx1

IRET ; конец прерывания

SUBR\_INT ENDP

MAIN PROC FAR

PUSH DS

SUB AX, AX

PUSH AX

MOV AX, DATA

MOV DS, AX

MOV AH, 35H ; функция получения вектора

MOV AL, 60h ; номер вектора

INT 21H

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

INT 60h

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
```