

**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.**

Студент гр. 9382

Юрьев С.Ю.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Познакомиться с прерываниями, изучить их работу, разработать собственное прерывание.

### **Задание:**

Вариант 4А:

4: Номер и назначение заменяемого вектора прерывания - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

А: Действие, реализуемое программой обработки прерывания — печать сообщения на экране

### **Выполнение работы:**

В ходе работы была реализована программа на языке Ассемблер, которая сохраняет старый вектор прерывания, устанавливает новый, вызывает его обработку и восстанавливает старый вектор прерывания.

В программе реализована отдельная процедура MY\_INT, которая является программой обработки прерывания. В ней, в соответствии с вариантом реализуется вывод строки «Hello world!» на экран.

Также используются следующие инструкции:

- OUT - выводит данные из регистра AL или AX (ИСТОЧНИК) в порт ввода-вывода..
- IRET - interrupt return - возврат после обработки прерывания.

В процедуре MAIN происходит сохранение старого, установка нового и восстановление старого векторов прерывания. Используемые функции и инструкции:

- 35h - возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX. 3

- 25h - помещает заданные адреса сегмента и смещения обработчика в вектор прерывания с заданным номером.
- CLI - сбрасывает флаг прерывания (флаг IF). Выполнение этой команды отключает аппаратные прерывания.
- STI - устанавливает флаг разрешения прерываний IF в регистре флагов, разрешая все аппаратные прерывания.

Исходный код см. в приложении А.

### **Выводы.**

Было произведено знакомство с прерываниями, была изучена их работа, было разработано собственное прерывание.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
AStack SEGMENT STACK
    DB 1024 DUP(?)
AStack ENDS
```

```
DATA SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0
DATA ENDS
```

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

```
MY_INT PROC FAR
JMP BEGIN
TEXT DB 'Hello world!$'
KEEP_SS DW 0
KEEP_SP DW 0
KEEP_AX DW 0
INTSTACK DW 16 DUP(?)
BEGIN:
MOV KEEP_SP, SP
MOV KEEP_AX, AX
MOV AX, SS
MOV KEEP_SS, AX
MOV AX, KEEP_AX
MOV SP, OFFSET BEGIN
MOV AX, SEG INTSTACK
MOV SS, AX
PUSH AX
PUSH DX
PUSH DS
MOV AX, SEG MY_INT
MOV DS, AX
MOV AH, 9H
MOV DX, OFFSET TEXT
INT 21H
POP DS
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
MOV AH, 35H
MOV AL, 08H
INT 21H
MOV KEEP_IP, BX
MOV KEEP_CS, ES
PUSH DS
MOV DX, OFFSET MY_INT
MOV AX, SEG MY_INT
MOV DS, AX
MOV AH, 25H
MOV AL, 08H
INT 21H
POP DS
MY_LOOP:
MOV AH, 01H
INT 21H
CMP AL, 13
JE MY_END
JMP MY_LOOP
MY_END:
CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 08H
INT 21H
POP DS
STI
MOV AH, 4CH
INT 21H
MAIN ENDP
CODE ENDS
END MAIN
```