

**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. 9382

\_\_\_\_\_

Субботин М.О.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

## **Цель работы.**

Научиться разрабатывать собственное прерывание

## **Основные теоретические положения.**

Прерывание - это процесс вызова процедур для выполнения некоторой задачи, обычно связанной с обслуживанием некоторых устройств (обработка сигнала таймера, нажатия клавиши и т.д.). Когда возникает прерывание, процессор прекращает выполнение текущей программы (если ее приоритет ниже) и запоминает в стеке вместе с регистром флагов адрес возврата(CS:IP) - места, с которого будет продолжена прерванная программа. Затем в CS:IP загружается адрес программы обработки прерывания и ей передается управление. Адреса 256 программ обработки прерываний, так называемые векторы прерывания, имеют длину по 4 байта (в первых двух хранится значение IP , во вторых - CS) и хранятся в младших 1024 байтах памяти. Программа обработки прерывания должна заканчиваться инструкцией IRET (возврат из прерывания), по которой из стека восстанавливается адрес возврата и регистр флагов.

Программа обработки прерывания - это отдельная процедура, имеющая структуру:

```
SUBR_INT PROC FAR  
PUSH AX ; сохранение изменяемых регистров
```

...

```
<действия по обработке прерывания> POP AX ; восстановление  
регистров
```

...

```
MOV AL,20H OUT 20H,AL IRET
```

```
SUBR_INT ENDP
```

Две последние строки обработчика прерывания, указанные перед командой IRET выхода из прерывания, необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное.

Замечание: в лабораторной работе действиями по обработке прерывания может быть вывод на экран некоторого текста, вставка цикла задержки в вывод сообщения или включение звукового сигнала.

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. В соответствии с этим, программа должна содержать следующие инструкции:

```
; -- в сегменте данных
KEEP_CS DW 0 ; для хранения сегмента
KEEP_IP DW 0 ; и смещения вектора прерывания
```

```
; -- в начале программы
MOV AH, 35H ; функция получения вектора
MOV AL, 1CH ; номер вектора
INT 21H
MOV KEEP_IP, BX ; запоминание смещения
MOV KEEP_CS, ES ; и сегмента вектора прерывания
```

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.

```
PUSH DS
MOV DX, OFFSET ROUT ; смещение для процедуры в DX
MOV AX, SEG ROUT ; сегмент процедуры
```

```
MOV DS, AX
MOV AH, 25H
MOV AL, 60H
INT 21H
```

```
; помещаем в DS
; функция установки вектора
```

```
; номер вектора
; меняем прерывание
```

```
POP DS
```

Далее может выполняться вызов нового обработчика прерывания.

В конце программы восстанавливается старый вектор прерывания CLI

```
PUSH DS
MOV DX, KEEP_IP
```

```
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 1CH
```

```
INT 21H ; восстанавливаем старый вектор прерывания
POP DS
```

STI

### **Ход выполнения:**

Задача состоит в том, чтобы прерывание 1СН переопределить на вывод строки.

Были созданы 2 переменные – в которых записаны строки.

Программа для обработки прерываний представляет собой процедуру WRITE\_SOME.

Переопределенное прерывание вызывается после сообщения в DX сдвиг переменной GREETING представляющую строку, а также вызывается после сдвига HELLOWORLD.

Остальная структура программы соответствует указаниям из методических материалов.

### **Исходный код программы:**

```
STACKSG SEGMENT PARA STACK 'Stack'
    DW    1024 DUP(?)
STACKSG  ENDS

DATASG SEGMENT PARA 'Data'                                ;SEG DATA
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения вектора прерывания
    GREETING DB 'Subbotin Maksim 9382 $'
    HELLOWORLD DB 'Hello World!!! $'
DATASG    ENDS                                           ;ENDS DATA

CODE    SEGMENT                                           ;SEG CODE
ASSUME DS:DataSG, CS:Code, SS:STACKSG

WRITE_SOME PROC FAR
    PUSH AX    ; сохранение изменяемых регистров
    ;<действия по обработке прерывания>
    mov  AH,9 ;вызов того,
    int  21h ;что хранится в dx
```

```

    POP AX    ;восстановление регистров
    MOV AL, 20H
    OUT 20H,AL
    IRET
WRITE_SOME ENDP

```

```

Main    PROC FAR

```

```

    mov ax, DATASG          ;ds setup
    mov ds, ax

```

```

    MOV AH, 35H    ; функция получения вектора
    MOV AL, 1CH    ; номер вектора
    INT 21H
    MOV KEEP_IP, BX ; запоминание смещения
    MOV KEEP_CS, ES ; и сегмента вектора прерывания

```

```

    PUSH DS
    MOV DX, OFFSET WRITE_SOME ; смещение для процедуры в DX
    MOV AX, SEG WRITE_SOME    ; сегмент процедуры
    MOV DS, AX                ; помещаем в DS
    MOV AH, 25H                ; функция установки вектора
    MOV AL, 1CH                ; номер вектора
    INT 21H                    ; меняем прерывание
    POP DS

```

```

    mov  DX, OFFSET GREETING ;так как наше переопределенное прерывние
выводит строку, запишем в dx то, что надо вывести

```

```

    int 1Ch ;вызываем наше переопределенное прерывание

```

```

    mov  DX, OFFSET HELLOWORLD
    int 1Ch

```

```

CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 1CH
INT 21H      ; восстанавливаем старый вектор прерывания
POP DS
STI

mov ah,4Ch;
int 21h;

```

```

Main    ENDP
CODE    ENDS
END Main                                ;ENDS CODE

```

### **Тестирование.**

При переменных

```

GREETING DB 'Subbotin Maksim 9382 $'
HELLOWORLD DB 'Hello World!!! $'

```

Вызывается прерывания 1CH

Программа выводит:

```

Subbotin Maksim 9382 Hello World!!!

```

Что явно и ожидается от такого переопределенного прерывания.

**Выводы.**

Я научился разрабатывать собственное прерывание.

**Ответы на вопросы.**