

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

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

Студент гр. 1303

Жилин И.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить особенности прерываний на языке Ассемблера, написать собственное прерывание.

Задание.

Вариант 2А.

Написать прерывание 60h – прерывание пользователя, должно генерироваться в программе. Выполнить вывод сообщения на экран заданное число раз, после чего вставить фиксированную задержку и вывести сообщение о завершении обработчика.

Выполнение работы.

Объявляются два двухбайтовых сегмента памяти KEEP_IP и KEEP_CS. Они используются для сохранения смещения до оригинального прерывания и позволяют в дальнейшем восстановить исходные вектора прерывания. Также строка с сообщением для вывода, строка с сообщением о завершении обработчика и переменная, содержащая заданное число вывода.

Описывается процедура SUBR_INT, которая является написанным пользовательским прерыванием. В данной процедуре в начале все регистры, которые будут изменены, для сохранения кладутся в стек, затем осуществляется вывод сообщения заданное количество раз, задержка в одну секунду и вывод сообщения о завершении.

После выполнения вышеописанных операций все сохраненные регистры изымаются из стека, а также обеспечивается разрешение прерываний более низкого уровня во время действия данного.

В главной процедуре смещение и сегмент прерывания, которое требуется заменить, сохраняются в объявленные сегменты памяти. Функция 35H прерывания 21H дает вектор прерывания, записанного в нижний байт регистра AX. Смещение и сегмент данного регистра записываются в регистры BX и ES,

соответственно, они сохраняются в KEEPR_IP и KEEPR_CS. Далее записывается новое прерывание.

Функция 25H считывает смещение до него из DX и сегмент из DS и устанавливает его в вектор прерывания. Исходный вектор прерывания восстанавливается, и программа завершается. Исходный код программы см. в приложении А.

Вывод.

В ходе выполнения лабораторной работы были изучены виды прерываний и работа с ними. В соответствии с заданием было разработано собственное прерывание.

ПРИЛОЖЕНИЕ А

Тексты исходных файлов программ L5.asm.

```
DATA SEGMENT
```

```
    KEEP_CS dw 0
```

```
    KEEP_IP dw 0
```

```
    MyString db 10, 13, 'TLOU$'
```

```
    OutNum dw 3
```

```
    EndMessage db 10, 13, 'End int$'
```

```
DATA ENDS
```

```
AStack SEGMENT STACK
```

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

```
AStack ENDS
```

```
CODE SEGMENT
```

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

```
SUBR_INT PROC FAR
```

```
    push ax
```

```
    push cx
```

```
    push bx
```

```
    push dx
```

```
    mov cx, OutNum
```

```
str_out:
```

```
    cmp cx, 0
```

```
    je stop_output
```

```
    mov ah, 9
```

```
    mov dx, offset MyString
```

```
    int 21h
```

```
    dec cx
```

```
    jmp str_out
```

```

    stop_output:

    mov cx, 0fh
    mov dx, 4240h
    mov ah, 86h
    int 15h

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

    pop dx
    pop bx
    pop cx
    pop ax
    mov al, 20h
    out 20h, al
    iret
SUBR_INT ENDP

```

```

Main PROC FAR
    push DS
    sub 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

        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

        mov ah, 4ch
        int 21h
Main      ENDP
CODE ENDS
        END Main

```