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

ОТЧЕТ

по лабораторной работе №5
по дисциплине «ОргЭВМиС»
Тема: Написание собственного прерывания.

Студент гр. 0382 Гудов Н.Р.

Преподаватель Ефремов М.А.

Санкт-Петербург
2021

Цель работы.

Изучение прерывания на языке Ассемблера. Написать собственное прерывание.

Задание.

Вар 21.

Прерывание 23h + Выполнить ввод и печать заданного количества символов, после чего вывести сообщение о завершении обработчика.

Выполнение работы.

В главной процедуре main сначала вызывается функция 35h прерывания 21h для получения текущего вектора прерывания 23h, который генерируется по нажатию Control+C. Значения CS этого вектора, хранящегося в результате в ES, и IP, хранящегося в BX, записываются в память для того, чтобы вернуть этот вектор в конце программы. Для задания нового адреса прерывания используется функция 23h прерывания 21h. Перед которой в DX записывается смещение процедуры с созданным прерыванием, а сегмент записывается в DS, в AL записывается номер прерывания. Далее программа считывает ввод с клавиатуры с помощью функции 00h прерывания 16h: при нажатии Control+C предлагается ввести 5 символов, которые тут же печатаются вместе с сообщением об окончании прерывания. В конце программы с помощью функции 25h и сохраненных CS, IP восстанавливается изначальный вектор для прерывания.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	fwaaw	fwaaw int end	ok
2.	drhhj	drhhj int end	ok
3.	24dr!	24dr! int end	ok

Выводы.

Были изучены прерывания на языке Ассемблера. Создано собственное прерывание.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

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

STACK SEGMENT STACK
        DW 1024 DUP(?)
STACK ENDS

DATA SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0
    string db 0dh,0ah,'int end',0dh,0ah,'$'
DATA ENDS

CODE SEGMENT
    Interruption PROC FAR
        jmp int_start
        SAVE_SS dw 0
        SAVE_SP dw 0
        int_stack dw 32 dup(0)

    int_start:
        mov SAVE_SS, SS
        mov SAVE_SP, SP
        mov ax, seg int_stack
        mov SS, SP
        mov SP, offset int_start
        push ax
        push dx

      push bx
      push ds
      push cx

      mov cx, 0
    input:
```

```
 mov ah,01h
    int 21h
 inc cx
 cmp cx, 5
 jne input

prnt:
 mov bx, DATA
 mov ds, bx

    mov ah, 09h
    mov dx, offset string
    int 21h

    pop ax
    pop dx

 pop bx
 pop ds
 pop cx

    pop cx
    mov SS, SAVE_SS
    mov SP, SAVE_SP
    mov AL, 20h
    out 20h, AL

IRET
Interruption ENDP

Main PROC FAR
    mov ah, 35h
    mov al, 23h
    int 21h
    mov KEEP_IP, bx
    mov KEEP_CS, es

    push ds
    mov dx, offset Interruption
```

```
        mov ax, seg Interruption
        mov ds, ax
        mov ah, 25h
        mov al, 23h
        int 21h
        pop ds

        begin:
                mov ah,0
                int 16h
                cmp al, 'q'
                je quit
                cmp al,3
                jnz begin

                int 23h
                jmp begin

           quit:

        CLI
        push ds
        mov  dx, KEEP_IP
        mov  ax, KEEP_CS
        mov  ds, ax
        mov  ah, 25h
        mov  al, 23h
        int 21h
        pop ds
        STI

        mov ah, 4ch
           int 21h

    Main ENDP
CODE ENDS
END Main
```

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab5.lst

```
Microsoft (R) Macro Assembler Version 5.10                  12/18/21
18:58:0
                                                             Page
1-1


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

0000                    STACK SEGMENT STACK
0000  0400[                        DW 1024 DUP(?)
       ????
                ]

0800                    STACK ENDS

0000                    DATA SEGMENT
0000  0000                    KEEP_CS DW 0
0002  0000                    KEEP_IP DW 0
0004  0D 0A 69 6E 74 20       string        db        0dh,0ah,'int
end',0dh,0ah,'$'
      65 6E 64 0D 0A 24
0010                    DATA ENDS

0000                    CODE SEGMENT
0000                        Interruption PROC FAR
0000  EB 45 90                   jmp int_start
0003  0000                       SAVE_SS dw 0
0005  0000                       SAVE_SP dw 0
0007  0020[                      int_stack dw 32 dup(0)
       0000
                ]


0047                        int_start:
0047  2E: 8C 16 0003 R                mov SAVE_SS, SS
004C  2E: 89 26 0005 R                mov SAVE_SP, SP
```

```
0051  B8 ---- R             mov ax, seg int_stack
0054  8E D4                      mov SS, SP
0056  BC 0047 R             mov SP, offset int_start
0059  50                    push ax
005A  52                    push dx

005B  53                  push bx
005C  1E                  push ds
005D  51                  push cx

005E  B9 0000                   mov cx, 0
0061                     input:

0061  B4 01                     mov ah,01h
0063  CD 21                       int 21h
0065  41                   inc cx
0066  83 F9 05                  cmp cx, 5
0069  75 F6                    jne input

006B                     prnt:
006B  BB ---- R           mov bx, DATA
006E  8E DB                     mov ds, bx

0070  B4 09                       mov ah, 09h
0072  BA 0004 R             mov dx, offset string
```

```
0075  CD 21                    int 21h

0077  58                  pop ax
0078  5A                  pop dx

0079  5B                pop bx
007A  1F                pop ds
007B  59                pop cx

007C  59                  pop cx
007D  2E: 8E 16 0003 R              mov SS, SAVE_SS
0082  2E: 8B 26 0005 R              mov SP, SAVE_SP
0087  B0 20                   mov AL, 20h
0089  E6 20                   out 20h, AL

008B  CF              IRET
008C                  Interruption ENDP

008C                  Main PROC FAR
008C  B4 35                   mov ah, 35h
008E  B0 23                   mov al, 23h
0090  CD 21                   int 21h
0092  89 1E 0002 R            mov KEEP_IP, bx
0096  8C 06 0000 R            mov KEEP_CS, es

009A  1E                  push ds
009B  BA 0000 R           mov dx, offset Interruption
009E  B8 ---- R           mov ax, seg Interruption
00A1  8E D8                   mov ds, ax
00A3  B4 25                   mov ah, 25h
00A5  B0 23                   mov al, 23h
00A7  CD 21                   int 21h
00A9  1F                  pop ds
```

```
00AA                        begin:
00AA  B4 00                             mov ah,0
00AC  CD 16                             int 16h
00AE  3C 71                             cmp al, 'q'
00B0  74 08                             je quit
00B2  3C 03                             cmp al,3
00B4  75 F4                             jnz begin

00B6  CD 23                             int 23h
00B8  EB F0                             jmp begin

00BA                           quit:

00BA  FA                    CLI
00BB  1E                    push ds
00BC  8B 16 0002 R                  mov  dx, KEEP_IP
00C0  A1 0000 R             mov  ax, KEEP_CS
00C3  8E D8                         mov  ds, ax
00C5  B4 25                         mov  ah, 25h
00C7  B0 23                         mov  al, 23h
```

```
00C9  CD 21                    int 21h
00CB  1F                  pop ds
00CC  FB                  STI

00CD  B4 4C                    mov ah, 4ch
00CF  CD 21                       int 21h

00D1                      Main ENDP
00D1                CODE ENDS
                    END Main
```

Segments and Groups:

N a m e	Length	Align	Combine Class
CODE	00D1	PARA	NONE
DATA	0010	PARA	NONE
STACK	0800	PARA	STACK

Symbols:

N a m e	Type	Value	Attr
BEGIN	L NEAR	00AA	CODE
INPUT	L NEAR	0061	CODE
INTERRUPTION	F PROC	0000	CODE Length = 008C
INT_STACK	L WORD	0007	CODE Length = 0020
INT_START	L NEAR	0047	CODE
KEEP_CS	L WORD	0000	DATA
KEEP_IP	L WORD	0002	DATA
MAIN	F PROC	008C	CODE Length = 0045
PRNT	L NEAR	006B	CODE
QUIT	L NEAR	00BA	CODE
SAVE_SP	L WORD	0005	CODE
SAVE_SS	L WORD	0003	CODE

```
STRING . . . . . .     L BYTE    0004  DATA

@CPU . . . . . .       TEXT  0101h
@FILENAME  . . . . .   TEXT  lab5
@VERSION . . . . . .   TEXT  510


    111 Source  Lines
    111 Total   Lines
     21 Symbols

  48016 + 461291 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```