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: Написание собственного прерывания
Вариант 13

Студент гр.1381 _____ Луценко Д.А.

Преподаватель _____ Ефремов М. А.

Санкт-Петербург
2022

Цель работы.

Реализовать собственное прерывание на языке Ассемблера

Задание.

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

f – Вывод на экран заданного количества (3-5) сообщений, задержка между которыми возрастает в 2 раза, начиная от 1 сек.

Выполнение работы.

В сегменте стека stack выделяется 1 Кбайт памяти, то есть *DW 512*.

В сегменте данных data содержится две переменных для хранения старого прерывания– prev_cs , prev_ip.

В сегменте кода определяется процедура func. Нынешнее состояние регистров сохраняется в стек, и восстанавливаются в конце процедуры. Далее, программа заходит во вложенный цикл. В первом цикле происходит печать сообщения в консоль. Во втором цикле происходит вызов процедуры, которая вызывает задержку. Длительность задержи меняется после 2 цикла при помощи логического сдвига влево. Также в первом цикле есть проверка на вывод сообщений, и если мы вывели необходимое кол-во сообщений, то переходим в конец процедуры и восстанавливаем регистры из стека.

В процедуре main. Записывается в ds смещение до data. Мы передаём в ah функцию установки вектора(25h) и в al передаём номер вектора(60h) и вызываем прерывание 21h и получаем наше прерывание. Значения регистров сохраняются в переменные. Новое прерывание func записывается в прерывание 60h, с помощью прерывания 21h. Далее, происходит вызов прерывания 60h. После вызова нового прерывания происходит восстановление старого прерывания и выход из программы.

Выводы.

В ходе выполнения лабораторной работы были изучены разные прерывания и работа с ними, было реализована собственное прерывание, а также написана программа в соответствии данному заданию.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: *lb5.asm*

```
AStack  SEGMENT STACK
    DB 512 DUP(?)
AStack  ENDS

DATA    SEGMENT
    prev_cs DW 0
    prev_ip DW 0
      MESSAGE DB 'Message', 0dh, 0ah, '$'
    MESSAGE_DELAY DB 'Delay...', 0dh, 0ah, '$'
DATA    ENDS

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


delay  PROC  NEAR
          push ax
          push bx
          push cx
          push dx

          mov ah,9h ;функция установки вектора
          mov dx,offset MESSAGE_DELAY ;в dx загружаем адрес сообщения
Message2
          int 21h ;вывод строки на экран

          pop dx
          pop cx
          pop bx
          pop ax
          ret
delay ENDP

Write  PROC  NEAR
          push ax
          push bx
          push cx
          push dx

          mov ah, 9h ;функция установки вектора
          int 21h  ;вывод строки на экран

          pop dx
          pop cx
          pop bx
          pop ax

          ret
Write  ENDP

FUNC PROC FAR
           push ax
```

```
        push bx
           push cx
           push dx


        mov dx, OFFSET MESSAGE
        mov cx, 4
        mov ax, 1
        lp1:
           mov dx, OFFSET MESSAGE
           mov bx, cx
           call Write
           cmp cx, 1
           je lp3
           mov cx, ax

           lp2:
               call delay
           loop lp2

        shl ax, 1
        mov cx,bx
        loop lp1

           lp3:
           pop dx
           pop cx
        pop bx
           pop ax
           mov al, 20h
           out 20h, al
           iret
FUNC ENDP

MAIN PROC FAR
    push ds
      sub ax, ax
      push ax
    mov ax, DATA
    mov ds, ax

    mov ah,25h ;функция установки вектора
    mov al,60h ;номер вектора
    int 21h
    mov prev_ip, bx
    mov prev_cs, es

    push ds
    mov dx, OFFSET FUNC
    mov ax, SEG FUNC
    mov ds, ax
    mov ah, 25h
    mov al, 60h
    int 21h
    pop ds
      int 60h
    CLI
    push ds
```

```
    mov dx, prev_ip
    mov ax, prev_cs
    mov ds, ax
    mov ah, 25h
    mov al, 60h
    int 21h
    pop ds
    STI
     ret

MAIN ENDP
CODE ENDS
      END MAIN
```

ПРИЛОЖЕНИЕ Б

ЛИСТИНГ ПРОГРАММЫ

Название файла: *lb5.lst*

```
__Microsoft (R) Macro Assembler Version 5.10                 11/21/22
23:34:1
                                                             Page     1-1


0000                    AStack  SEGMENT STACK
0000  0200[                      DB 512 DUP(?)
       ??
                ]

0200                    AStack  ENDS

0000                    DATA    SEGMENT
0000  0000                      prev_cs DW 0
0002  0000                      prev_ip DW 0
0004  4D 65 73 73 61 67          MESSAGE DB 'Message', 0dh, 0ah, '$'
      65 0D 0A 24
000E  44 65 6C 61 79 2E         MESSAGE_DELAY DB 'Delay...', 0dh, 0ah, '$'
      2E 2E 0D 0A 24
0019                    DATA    ENDS

0000                    CODE    SEGMENT
                            ASSUME CS:CODE, DS:DATA, SS:AStack


0000                    delay  PROC  NEAR
0000  50                        push ax
0001  53                        push bx
0002  51                        push cx
0003  52                        push dx

0004  B4 09                          mov ah,9h ;функция установки вектора
0006  BA 000E R                 mov dx,offset MESSAGE_DELAY ;в dx заг
                        ружаем адрес сообщения Message2
0009  CD 21                          int 21h ;вывод строки на экран

000B  5A                        pop dx
000C  59                        pop cx
000D  5B                        pop bx
000E  58                        pop ax
000F  C3                        ret
0010                    delay ENDP

0010                    Write  PROC  NEAR
0010  50                        push ax
0011  53                        push bx
0012  51                        push cx
0013  52                        push dx

0014  B4 09                          mov ah, 9h ;функция установки вектора
0016  CD 21                          int 21h  ;вывод строки на экран
```

```
0018  5A                        pop dx
0019  59                        pop cx
001A  5B                        pop bx
001B  58                        pop ax

001C  C3                        ret
001D                  Write  ENDP
__Microsoft (R) Macro Assembler Version 5.10                 11/21/22
23:34:1
                                                             Page     1-2


001D                  FUNC PROC FAR
001D  50                          push ax
001E  53                        push bx
001F  51                          push cx
0020  52                          push dx


0021  BA 0004 R                 mov dx, OFFSET MESSAGE
0024  B9 0004                       mov cx, 4
0027  B8 0001                       mov ax, 1
002A                            lp1:
002A  BA 0004 R                    mov dx, OFFSET MESSAGE
002D  8B D9                             mov bx, cx
002F  E8 0010 R                    call Write
0032  83 F9 01                          cmp cx, 1
0035  74 0D                             je lp3
0037  8B C8                             mov cx, ax

0039                               lp2:
0039  E8 0000 R                        call delay
003C  E2 FB                             loop lp2

003E  D1 E0                           shl ax, 1
0040  8B CB                           mov cx,bx
0042  E2 E6                           loop lp1

0044                               lp3:
0044  5A                           pop dx
0045  59                           pop cx
0046  5B                         pop bx
0047  58                           pop ax
0048  B0 20                            mov al, 20h
004A  E6 20                            out 20h, al
004C  CF                           iret
004D                  FUNC ENDP

004D                  MAIN PROC FAR
004D  1E                      push ds
004E  2B C0                           sub ax, ax
0050  50                       push ax
0051  B8 ---- R               mov ax, DATA
0054  8E D8                         mov ds, ax

0056  B4 25                         mov ah,25h ;функция установки вектора
```

```
 0058  B0 60                    mov al,60h ;номер вектора
 005A  CD 21                    int 21h
 005C  89 1E 0002 R             mov prev_ip, bx
 0060  8C 06 0000 R             mov prev_cs, es

 0064  1E                 push ds
 0065  BA 001D R          mov dx, OFFSET FUNC
 0068  B8 ---- R          mov ax, SEG FUNC
 006B  8E D8                    mov ds, ax
__Microsoft (R) Macro Assembler Version 5.10                 11/21/22
23:34:1
                                                            Page     1-3


 006D  B4 25                    mov ah, 25h
 006F  B0 60                    mov al, 60h
 0071  CD 21                    int 21h
 0073  1F                 pop ds
 0074  CD 60                      int 60h
 0076  FA                 CLI
 0077  1E                 push ds
 0078  8B 16 0002 R             mov dx, prev_ip
 007C  A1 0000 R          mov ax, prev_cs
 007F  8E D8                    mov ds, ax
 0081  B4 25                    mov ah, 25h
 0083  B0 60                    mov al, 60h
 0085  CD 21                    int 21h
 0087  1F                 pop ds
 0088  FB                 STI
 0089  CB                  ret

 008A                MAIN ENDP
 008A                CODE ENDS
                          END MAIN
__Microsoft (R) Macro Assembler Version 5.10                 11/21/22
23:34:1
                                                            Symbols-1


Segments and Groups:

                N a m e         Length      Align     Combine Class

ASTACK . . . . . .              0200  PARA  STACK
CODE . . . . . .                008A  PARA  NONE
DATA . . . . . .                0019  PARA  NONE

Symbols:

                N a m e         Type   Value      Attr

DELAY  . . . . . .              N PROC     0000  CODE  Length = 0010

FUNC . . . . . .                F PROC     001D  CODE  Length = 0030

LP1  . . . . . .                L NEAR     002A  CODE
LP2  . . . . . .                L NEAR     0039  CODE
LP3  . . . . . .                L NEAR     0044  CODE
```

```
MAIN . . . . . .      F PROC     004D  CODE  Length = 003D
MESSAGE  . . . . . .  L BYTE     0004  DATA
MESSAGE_DELAY  . . . . . .  L BYTE     000E  DATA

PREV_CS  . . . . . .  L WORD     0000  DATA
PREV_IP  . . . . . .  L WORD     0002  DATA

WRITE  . . . . . .    N PROC     0010  CODE  Length = 000D

@CPU . . . . . .      TEXT  0101h
@FILENAME  . . . . . .  TEXT  LB5
@VERSION . . . . . .  TEXT  510


    122 Source  Lines
    122 Total   Lines
     19 Symbols

  48030 + 459230 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```