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 Кбайтпамяти, то есть *DW 512*.

В сегменте данных data содержится две переменных для хранениястарого 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 cx,0fh; задаём начало временного промежутка в микросекундах
          mov dx,4240h; окончание временного промежутка
          mov ah,86h; установка вектора
          int 15h; вызов прерывания
          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                 11/28/22
23:31:5
                                                             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  B9 000F                          mov cx,0fh; задаём начало временного
                        промежутка в микросекундах
 0007  BA 4240                          mov dx,4240h; окончание временного пр
                        омежутка
 000A  B4 86                            mov ah,86h; установка вектора
 000C  CD 15                            int 15h; вызов прерывания
 000E  B4 09                            mov ah,9h ;функция установки вектора
 0010  BA 000E R                   mov dx,offset MESSAGE_DELAY ;в dx заг
                        ружаем адрес сообщения Message2
 0013  CD 21                            int 21h ;вывод строки на экран

 0015  5A                        pop dx
 0016  59                        pop cx
 0017  5B                        pop bx
 0018  58                        pop ax
 0019  C3                        ret
 001A                   delay ENDP

 001A                   Write  PROC  NEAR
 001A  50                        push ax
```

```
 001B  53                       push bx
 001C  51                       push cx
 001D  52                       push dx

 001E  B4 09                         mov ah, 9h ;функция установки вектора
 0020  CD 21                         int 21h  ;вывод строки на экран

 0022  5A                       pop dx
__Microsoft (R) Macro Assembler Version 5.10                 11/28/22
23:31:5
                                                            Page     1-2


 0023  59                       pop cx
 0024  5B                       pop bx
 0025  58                       pop ax

 0026  C3                       ret
 0027                 Write  ENDP

 0027                 FUNC PROC FAR
 0027  50                        push ax
 0028  53                      push bx
 0029  51                        push cx
 002A  52                        push dx


 002B  BA 0004 R              mov dx, OFFSET MESSAGE
 002E  B9 0004                     mov cx, 4
 0031  B8 0001                     mov ax, 1
 0034                         lp1:
 0034  BA 0004 R                 mov dx, OFFSET MESSAGE
 0037  8B D9                          mov bx, cx
 0039  E8 001A R                 call Write
 003C  83 F9 01                       cmp cx, 1
 003F  74 0D                          je lp3
 0041  8B C8                          mov cx, ax

 0043                            lp2:
 0043  E8 0000 R                     call delay
 0046  E2 FB                          loop lp2

 0048  D1 E0                        shl ax, 1
 004A  8B CB                        mov cx,bx
 004C  E2 E6                        loop lp1

 004E                            lp3:
 004E  5A                        pop dx
 004F  59                        pop cx
 0050  5B                      pop bx
 0051  58                        pop ax
 0052  B0 20                          mov al, 20h
 0054  E6 20                          out 20h, al
 0056  CF                        iret
 0057                 FUNC ENDP

 0057                 MAIN PROC FAR
```

```
 0057  1E                 push ds
 0058  2B C0                   sub ax, ax
 005A  50                  push ax
 005B  B8 ---- R          mov ax, DATA
 005E  8E D8                   mov ds, ax

 0060  B4 25                   mov ah,25h ;функция установки вектора
 0062  B0 60                   mov al,60h ;номер вектора
 0064  CD 21                   int 21h
 0066  89 1E 0002 R            mov prev_ip, bx
__Microsoft (R) Macro Assembler Version 5.10                 11/28/22
23:31:5
                                                             Page     1-3


 006A  8C 06 0000 R            mov prev_cs, es

 006E  1E                 push ds
 006F  BA 0027 R          mov dx, OFFSET FUNC
 0072  B8 ---- R          mov ax, SEG FUNC
 0075  8E D8                   mov ds, ax
 0077  B4 25                   mov ah, 25h
 0079  B0 60                   mov al, 60h
 007B  CD 21                   int 21h
 007D  1F                 pop ds
 007E  CD 60                    int 60h
 0080  FA                 CLI
 0081  1E                 push ds
 0082  8B 16 0002 R            mov dx, prev_ip
 0086  A1 0000 R          mov ax, prev_cs
 0089  8E D8                   mov ds, ax
 008B  B4 25                   mov ah, 25h
 008D  B0 60                   mov al, 60h
 008F  CD 21                   int 21h
 0091  1F                 pop ds
 0092  FB                 STI
 0093  CB                  ret

 0094                 MAIN ENDP
 0094                 CODE ENDS
                           END MAIN
__Microsoft (R) Macro Assembler Version 5.10                 11/28/22
23:31:5
                                                             Symbols-1


Segments and Groups:

                N a m e         Length      Align     Combine Class

ASTACK . . . . . .              0200  PARA  STACK
CODE . . . . . .                0094  PARA  NONE
DATA . . . . . .                0019  PARA  NONE

Symbols:

                N a m e         Type   Value      Attr
```

```
DELAY  . . . . . .      N PROC     0000  CODE  Length = 001A

FUNC . . . . . .        F PROC     0027  CODE  Length = 0030

LP1  . . . . . .        L NEAR     0034  CODE
LP2  . . . . . .        L NEAR     0043  CODE
LP3  . . . . . .        L NEAR     004E  CODE

MAIN . . . . . .        F PROC     0057  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     001A  CODE  Length = 000D

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


    126 Source  Lines
    126 Total   Lines
     19 Symbols

  48030 + 459230 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```