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

ОТЧЕТ

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

Студент гр. 1383 _____ Депрейс А.С

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Изучение прерываний на языке ассемблера, разработка собственного обработчика прерываний.

Задание.

Вариант 07.

Заменить прерывание от системного таймера, которое генерируется автоматически операционной системой 18 раз в сек, на выполнение чтения и вывода на экран отсчета часов реального времени из памяти CMOS (в формате BCD).

Выполнение работы

В сегменте данных хранятся:

keep_cs – место для хранения сегмента прерывания.

keep_ip – место для хранинния смещения прерывания.

Inter_msg – строка хранящая текущее время.

Написана функция WriteMsg, которая выводит сообщение в консоль.

Написана функция Set_time_string, которая совершает деление значения в регистре al, остаток записывает в регистр ah. Затем к ah и al прибавляет код ‘0’.

Написан обработчик прерывания, который считывает время из "постоянных" (CMOS) часов реального времени, затем конвертирует данные в строку и выводит ее.

В основном теле программы при помощи функции 35 прерывания 21H сохраняется значение вектора прерывания в keep_cs и keep_ip, затем при помощи функции 25 прерывания 21H устанавливается вектор прерывания на наш реализованный обработчик прерываний. После этого программа ждет 10 секундс при помощи функции 86 прерывания 15h.

В конце программы восстанавливается старый вектор прерывания.

Приложение А.

Исходный код программы.

```
ASSUME CS:Code, SS:Stack, DS:Data

Stack    SEGMENT  STACK
         DW 1024 DUP(0)
Stack    ENDS

Data SEGMENT
   inter_msg DB ?,?,':',?,?,':',?,?,0DH,'$'
   keep_cs DW 0
   keep_ip DW 0
Data ENDS

Code SEGMENT

WriteMsg  PROC  NEAR
   mov   ah,9
   int   21h
   ret
WriteMsg  ENDP

Set_time_string PROC NEAR
   mov bl, 10h
   mov ah, 0
   div bl
   add al, '0'
   add ah, '0'
   ret
Set_time_string ENDP

my_int PROC FAR
   push ax
   push dx
   push cx
read_time:
```

```
    mov ah, 02H
    int 1AH

    mov al, ch
    call Set_time_string
    mov inter_msg, al
    mov inter_msg + 1, ah

    mov al, cl
    call Set_time_string
    mov inter_msg + 3, al
    mov inter_msg + 4, ah

    mov al,dh
    call Set_time_string
    mov inter_msg + 6, al
    mov inter_msg + 7, ah

    mov dx, OFFSET inter_msg
    call WriteMsg


end_inter:
    pop cx
    pop dx
    pop ax

    mov al, 20H
    out 20H, al
    iret
my_int ENDP


main PROC FAR
```

```
push ds
xor  ax, ax
push ax

mov  ax, Data
mov  ds, ax

mov ah, 35H ; функция получения вектора
mov al, 08H ; номер вектора
int 21H
mov keep_ip, BX ; запоминание смещения
mov keep_cs, ES ; и сегмента

push ds
mov dx, OFFSET my_int ; смещение для процедуры в DX
mov ax, SEG my_int ; сегмент процедуры
mov ds, ax ; помещаем в DS
mov ah, 25H ; функция установки вектора
mov al, 08H ; номер вектора
int 21H ; меняем прерывание
pop ds


xor  ax, ax
mov  ah, 86h
mov  cx, 98h
mov  dx, 9680h
int  15h


cli
push ds

mov  dx, keep_ip
mov  ax, keep_cs
```

```
    mov  ds, ax

    mov  ah, 25h
    mov  al, 08h
    int  21h

    pop  ds
    sti

    ret
main ENDP

Code ENDS

END main
```