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.

Студент гр. 1383 _____ Ермакова В.М.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Изучить особенности прерываний на языке Ассемблера, написать собственное прерывание.

Задание.

Написать прерывание 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек. Выполнить вывод сообщения на экран заданное число раз, после чего вставить фиксированную задержку и вывести сообщение о завершении обработчика. Отвести в программе под стек не менее 1К байт.

Ход работы.

В сегменте данных DATA содержится две переменных для хранения старого прерывания, содержавшегося по смещению 08h.

В сегменте Astack выделяется 1Кбайт памяти, то есть db 1024.

В сегменте кода сначала определяем процедуру для печати времени print_cmos и процедуру пользовательского прерывания FUNC. В процедуре FUNC сначала сохраняются в стеке значения регистров до входа в прерывание. Далее выполняется чтение системного времени в порядке:

- Год
- Месяц
- День
- Час
- Минута
- Секунда

Формат вывода: “<год>-<месяц>-<день> <часы>h <минуты>:<секунды>”.
Далее из стека извлекаются сохранённые значения регистров.

Вызов прерывания происходит в процедуре MAIN. В переменных, указанных в сегменте данных, сохраняется старое прерывание. Новое прерывание FUNC записывается по смещению 08h, с помощью функции 25h прерывания 21h.

Вывод.

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

Приложение А

Исходный код программы

Название файла: main50.asm

```
AStack  SEGMENT STACK
    DB 1024 DUP(?)
AStack  ENDS

DATA    SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0
DATA    ENDS

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

print_cmos proc near
        out         70h,al
        in          al,71h
        push        ax
        mov         cl, 4
        shr         al,cl
        add         al,'0'
        int         29h
        pop         ax
        and         al,0Fh
        add         al,30h
        int         29h
        ret
print_cmos endp

FUNC PROC FAR
            push ax
            push bx
            push cx
            push dx
            push ds
```

```
    mov        al,0Bh
out         70h,al
in          al,71h
and         al,11111011b
out         71h,al
mov         al,32h
call        print_cmos
mov         al,9
call        print_cmos
mov         al,'-'
int         29h
mov         al,8
call        print_cmos
mov         al,'-'
int         29h
mov         al,7
call        print_cmos
mov         al,' '
int         29h
mov         al,4
call        print_cmos
mov         al,'h'
int         29h
mov         al,' '
int         29h
mov         al,2
call        print_cmos
mov         al,':'
int         29h
mov         al,0h
call        print_cmos

    pop ds
    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, 35h
    mov al, 08h
    int 21h
    mov KEEP_IP, bx
    mov KEEP_CS, es

    push ds
    mov dx, OFFSET FUNC
    mov ax, SEG FUNC
    mov ds, ax
    mov ah, 25h
    mov al, 08h
    int 21h
    pop ds

      int 08h

      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
```

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