

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.
Вариант 19

Студентка гр. 1383

Седова Э.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить прерывания на языке Ассемблера и написать собственное прерывание.

Задание.

Разработать собственное прерывание:

3 - 23h - прерывание, генерируемое при нажатии клавиш Control+C; 4 - 16h - прерывание от клавиатуры (по заданному скан-коду клавиши делать действия A-G, указанные ниже).

Е - Выполнить чтение и вывод на экран отсчета часов реального времени из памяти CMOS (в формате BCD).

Выполнение работы.

В сегменте Astack выделяется 1Кбайт памяти, как и требуется в задании.

В сегменте кода определяется процедура пользовательского прерывания TIME. С помощью команды push в стек записываются значения регистров до входа в прерывание. Команда 2ch прерывания 21h возвращает текущее время, как оно известно системе. Далее, поочередно, с помощью команды call, вызываются процедуры hours, minutes и seconds, в которых вызывается процедура output. В этой процедуре производится вывод текущего времени на экран.

В процедуре MAIN вызывается функция 35h прерывания 21h, возвращающая значение вектора прерывания, лежащего по смещению 23h. Пользовательское прерывание TIME записывается в прерывание 23h. Далее происходит вызов данного прерывания в случае считывания с клавиатуры комбинации клавиш ctrl+C. После вызова прерывания происходит восстановление старого вектора прерывания и завершение программы.

Программный код см. в приложении А.

Вывод.

В ходе выполнения работы были изучены виды прерываний на языке Ассемблера. В соответствии с заданием было создано собственное прерывание.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 5lb.asm

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

DATA     SEGMENT
        KEEP_CS DW 0
        KEEP_IP DW 0
DATA     ENDS

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

hours proc near
        mov al, ch
        call output
        ret
hours endp

minutes proc near
        mov al, cl
        call output
        ret
minutes endp

seconds proc near
        mov al, dh
        call output
        ret
seconds endp

output proc near
        aam
        add ax, 3030h
        mov dl, ah
        mov dh, al
        mov ah, 02
        int 21h
        mov dl, dh
        int 21h
        ret
output endp

TIME PROC FAR
        push ax
        push cx
        push dx
        push bx
        mov ah, 2ch
        int 21h

        call hours
        mov ah, 2 ; В ы в о д ' : '
```

```

        mov dl, ':'
        int 21h
        call minutes
        mov ah, 2; ; В ы в о д ':'
        mov dl, ':'
        int 21h
        call seconds

        pop bx
        pop dx
        pop cx
        pop ax
        mov al, 20h
        out 20h, al
        iret
TIME ENDP

MAIN PROC FAR
    push ds
    sub ax, ax
    push ax
    mov ax, DATA
    mov ds, ax
    mov ax, 3523h
    int 21h
    mov KEEP_IP, bx
    mov KEEP_CS, es
    push ds
    mov dx, OFFSET TIME
    mov ax, SEG TIME
    mov ds, ax
    mov ah, 25h
    mov al, 23h
    int 21h
    pop ds
    read:
        mov ah, 0
        int 16h
        cmp al, 3
        jnz read
    int 23h

    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

```