

**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.**

Студентка гр. 0382

Охотникова Г.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Целью данной работы является написание собственного прерывания

### **Задание.**

Вариант 12:

2 — 60h — прерывание пользователя — должно генерироваться в программе.

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

### **Выполнение.**

Процедура *show\_time*, которая показывает время на экране:

Отправляем регистр *AL* в порт *70H*, затем выполняем чтение данных из порта *71H* для получения данных из *CMOS*. Выделяем старшие 4 бита регистра *AL*, добавляем ASCII-код нуля и вызываем *int 29h* для показа символа на экран. Затем выделяем младшие 4 бита регистра *AL*, снова добавляем ASCII-код нуля и вызываем *int 29h*.

Процедура *SUBR\_INT*, в которой реализовано пользовательское прерывание:

Контрольным регистром для работы с CMOS является регистр *B*. Затем происходит чтение данных из порта *71H*. С помощью логического и происходит обнуление второго бита (так как результат операции поместится в регистр *AL*). Затем мы снова помещаем данные в порт *70H*.

Помещаем в *AL 04H*, что отвечает за часы в CMOS. Вызываем процедуру *show\_time*. Затем в регистр *AL* записываем разделитель — двоеточие, вызываем функцию *int 29h* для показа символа на экран, записываем в *AL 02H*, что отвечает за минуты и снова вызываем процедуру *show\_time*. Аналогично происходит с секундами, только в *AL* мы помещаем *00H*.

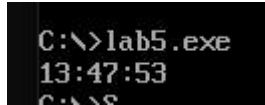
Затем происходит восстановление регистров и выход из прерывания.

В процедуре *MAIN* после вызова функции *int 35H* в регистрах *ES* и *BX* окажутся, соответственно, сегмент и смещение обработчика прерывания.

Данные значения запишем в *keep\_cs* и *keep\_ip*. После вызова функции *int 25H* происходит установка пользовательского прерывания и его вызов.

Исходный код программы см. в приложении А.

### **Тестирование.**



```
C:\>lab5.exe  
13:47:53  
C:\>
```

### **Выводы.**

При выполнении данной лабораторной работы было разработано собственное прерывание, которое выводит на экран время в формате BCD.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
DATA SEGMENT
    keep_cs dw 0
    keep_ip dw 0
DATA ENDS

AStack SEGMENT STACK
    DW 512 DUP(?)
AStack ENDS

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

.186                                ;for shr al, 4

show_time PROC NEAR
    out 70H, AL                    ;send AL in CMOS's port
    in AL, 71H                    ;read data
    push AX
    shr AL, 4                      ;select the highest 4 bits
    add AL, '0'                   ;add ASCII-code 0
    int 29H                       ;show symbol
    pop AX
    and AL, 0FH                   ;select the lower four bits
    add AL, 30H                   ;add ASCII-code 0
    int 29H                       ;show symbol
    ret
show_time ENDP

SUBR_INT PROC FAR
    jmp start
    KEEP_SS DW 0
    KEEP_SP DW 0
    KEEP_AX DW 0
    KEEP_STACK DW 512 DUP(?)

start:
    mov KEEP_SP, SP
    mov KEEP_AX, AX
    mov KEEP_SS, SS
    mov SP, OFFSET start
    mov AX, SEG KEEP_STACK
    mov SS, AX
    mov AX, KEEP_AX

    push AX                        ;save registers
    push DS

    mov AX, SEG SUBR_INT
    mov DS, AX
    mov AX, KEEP_AX

    mov AH, 29H
```

```

    mov AL, 0BH                ;control register B for CMOS
    out 70H, AL                ;port 70H is port of CMOS
    in AL, 71H                 ;port 71H is data of CMOS
    and AL, 11111011b          ;for BCD
    out 71H, AL                ;write back
    mov AL, 4                   ;CMOS 04h is hour
    call show_time

    mov AL, ':'                 ;separator
    int 29H
    mov AL, 2                   ;CMOS 02h is minutes

    call show_time

    mov AL, ':'                 ;separator
    int 29H
    mov AL, 0                   ;CMOS 00h is second

    call show_time

    mov AL, ' '                 ;separator
    int 29H

    pop DS
    pop AX                     ;recover registers
    mov SP, KEEP_SP
    mov AX, KEEP_SS
    mov SS, AX
    mov AX, KEEP_AX
    mov AL, 20H
    out 20H, AL
    IRET
    SUBR_INT ENDP

Main PROC FAR
push DS
    sub AX, AX
    push AX
    mov AX, DATA
    mov DS, AX

    mov AH, 35h
    mov AL, 60h
    int 21
    mov keep_ip, BX
    mov keep_cs, ES

    push DS
    mov DX, OFFSET SUBR_INT
    mov AX, SEG SUBR_INT
    mov DS, AX
    mov AH, 25H
    mov AL, 60H
    int 21H
    pop DS

```

```

int 60h                                ;call user's interupt

CLI
push DS
mov DX, KEEP_IP
mov AX, KEEP_CS
mov DS, AX
mov AH, 25H
mov AL, 60h
int 21H
pop DS
STI

ret

Main ENDP
CODE ENDS
END Main

```