

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. 0383

Куртова К. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Написать собственное прерывание.

Вариант — 5 (1е).

08h: Прерывание от системного таймера — генерируется автоматически операционной системой 18 раз в секунду.

Чтение и вывод на экран отсчёта часов реального времени из памяти CMOS (формат BSD).

Ход работы.

Пояснение по смыслу некоторых команд.

Процедура SUBR_INT (прерывание):

- 1) В начале процедуры сохраняются состояния регистров, в конце процедуры они восстанавливаются. Последовательность команд

```
mov al, 20h  
out 20h, al
```

обеспечивает разрешение работы прерываний с более низким приоритетом.

- 2) Для общения с CMOS и регистрами RTC выделяются порты ввода-вывода от 70h до 7Fh. В данном случае используются порты 70h и 71h.

Порт 70h существует для записи и предоставляет индекс для выбора регистра CMOS. Порт 71h необходим для доступа к данным CMOS.

- 3) В программе используется содержимое регистра 0Bh. Необходимо обнулить второй бит регистра, чтобы представлять время в двоично-десятичном формате (BSD).

- 4) Регистры 04h, 02h и 00h содержат информацию о текущем часе, минуте, секунде соответственно. Содержимое этих регистров запоминается в AL и выводится с помощью прерывания 29h и процедуры print_CMOS.

Процедура print_CMOS:

- 1) Из CMOS читаются данные, выделяются сначала 4 старших бита, которые выводятся на экран, после выделяются младшие 4 бита, которые также выводятся на экран.

Процедура Main:

- 1) Сохраняется адрес начала PSP.
- 2) Последовательностью команд, данных в указании к работе, устанавливается вектор прерывания с номером 08h.
- 3) Новое прерывание вызывается командой int 08h.
- 4) Восстанавливается изначальный вектор прерывания.

Тестирование программы.

Программа запускается вводом в командную строку lab5.exe. После этого на экране появляются часы реального времени в формате ...h ...m ...s (часы, минуты, секунды), которые печатаются ~18 раз в секунду.

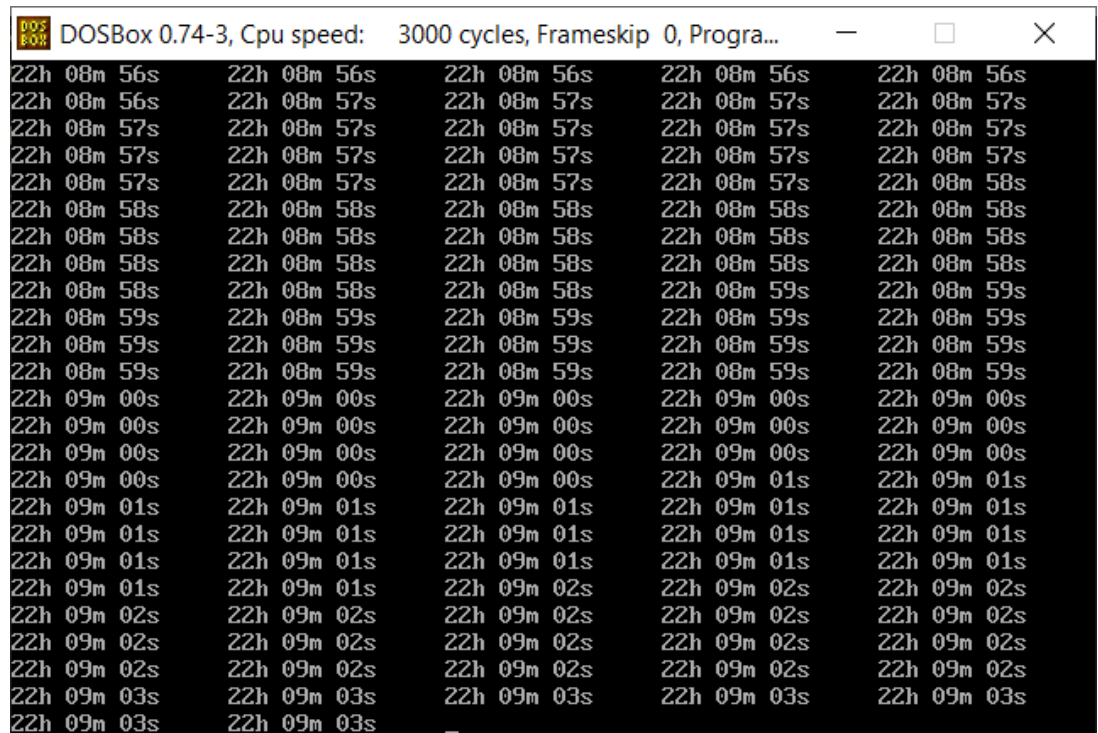


Рисунок 1 — Пример работы программы

Выводы.

В ходе данной лабораторной работы была изучена работа с прерываниями на языке ассемблера. Также было написано собственное прерывание.

ПРОТОКОЛ

Исходный код программы lab5.txt

```
; Задание 1e
; 08h - прерывание от системного таймера
; Чтение и вывод на экран отсчета часов реального
; времени из памяти CMOS (формат BSD)

ASSUME CS:CODE, DS:DATA, SS:STACK

STACK SEGMENT STACK
    DW 1024 DUP(?)
STACK ENDS

DATA SEGMENT
    KEEP_CS DW 0          ; Хранение сегмента
    KEEP_IP DW 0          ; Хранение приоритета
    KEEP_AX DW 0
    KEEP_SS DW 0
    KEEP_SP DW 0
DATA ENDS

CODE SEGMENT
.186
SUBR_INT PROC FAR
    start:
    mov  KEEP_AX, ax
    mov  KEEP_SS, ss
    mov  KEEP_SP, sp
    mov  sp, OFFSET start
    mov  ax, SEG STACK
    mov  ss, ax
    mov  ax, KEEP_AX

    push    ax
    push    ds
    mov  ax, SEG SUBR_INT
    mov  ds, ax
    mov  ax, KEEP_AX
    mov  ah, 29h          ; Функция вывода строки
    mov  al, 0Bh          ; CMOS: регистр состояния B
```

```

    out 70h, al          ; CMOS: индекс
    in  al, 71h          ; CMOS: данные
    and al, 11111011b    ; Обнуление 2-го бита
        out 71h, al      ; Запись бита обратно
        mov al, 4        ; CMOS: час
        call print_CMOS
        mov al, 'h'
        int 29h
        mov al, ' '
        int 29h
        mov al, 2        ; CMOS: минута
        call print_CMOS
        mov al, 'm'
        int 29h
    mov al, ' '
    int 29h
        mov al, 0        ; CMOS: секунда
        call print_CMOS
    mov al, 's'
    int 29h
    mov al, ' '
    int 29h

; Восстановление регистров
pop ds
pop ax
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

; Процедура печати времени из памяти CMOS
; Выводит на экран содержимое ячейки CMOS с номером в AL
print_CMOS PROC NEAR
    out 70h, al          ; CMOS: послать AL в индексный
порт
    in  al, 71h          ; CMOS: прочитать данные

```

```

        push    ax
        shr     al, 4
        add     al, '0'
        int     29h
        pop     ax
        and     al, 0Fh
        add     al, 30h
        int     29h
        ret
print_CMOS ENDP

; Головная процедура
Main PROC FAR
        ; Сохранение адреса начала PSP
        push    ds
        sub     ax, ax
        push    ax
        mov     ax, DATA
        mov     ds, ax

        ; Запоминание текущего вектора прерывания
        mov     ah, 35h                ; Возвращение текущего значения
вектора прерывания
        mov     al, 08h                ; Заносим номер вектора
        int     21h
        mov     KEEP_IP, bx            ; Запоминание смещения
        mov     KEEP_CS, es            ; Запоминание смещения

        ; Установка вектора прерывания
        push    ds
        mov     dx, OFFSET SUBR_INT    ; Смещение для процедуры в
DX
        mov     ax, SEG SUBR_INT        ; Сегмент процедуры
        mov     ds, ax                 ; Помещение в DS
        mov     ah, 25h                ; Функция установки вектора
        mov     al, 08h                ; Номер вектора
        int     21h                    ; Смена прерывания
        pop     ds

        ; Вызов прерывания
        int     08h

```

```

; Восстановление изначального вектора прерывания
cli                      ; Сброс IF
push     ds
mov  dx, KEEP_IP
mov  ax, KEEP_CS
mov  ds, ax
mov  ah, 25h
mov  al, 0h
int  21h                ; Восстановление вектора
pop  ds
sti                      ; Установка IF

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

```