

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

**ОТЧЕТ
по лабораторной работе №5
по дисциплине «ОргЭВМиС»
Тема: Написание собственного прерывания.**

Студент гр. 0382

Тюленев Т.В.

Преподаватели

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить работу с прерываниями, научиться создавать собственное прерывание.

Задание.

Вариант 3е: создать собственное прерывание с номером 23h, которое будет выдавать текущее время по нажатию Control+C.

Выполнение работы.

В главной процедуре **main** сначала вызывается функция 35h прерывания 21h для получения текущего вектора прерывания 23h, который генерируется по нажатию Control+C. Значения CS этого вектора, хранящегося в результате в ES, и IP, хранящегося в BX, записываются в память для того, чтобы вернуть этот вектор в конце программы. Для задания нового адреса прерывания используется функция 25h прерывания 21h, перед которой в DX записывается смещение процедуры с созданным прерыванием, а сегмент записывается в DS, в AL записывается номер прерывания. Далее программа считывает ввод с клавиатуры с помощью функции 00h прерывания 16h: при нажатии на Control+C выводится текущее время. А при нажатии на «q» программа завершается. В конце программы с помощью той же функции 25h и сохранённых CS и IP восстанавливается изначальный вектор для прерывания. Само прерывание реализовано в процедуре **inter**. В начале и в конце происходит сохранения и восстановления регистров, которые используются в процессе.

В созданном прерывании в AL записываются поочередно часы, минуты, секунды с помощью 04h, 02h, 00h соответственно и каждый раз с помощью функции `print_cmos` выводится на экран с добавлением разделяющего знака «:».

Тестирование.

При нажатии на Control+C выводится текущее время в формате чч:мм:сс.

При нажатии на Control+(любая клавиша, кроме «C») - ничего не выводится.

При нажатии на любую клавишу, кроме «C» - программа завершается.

Выводы.

Была изучена работа с прерываниями в Ассемблере и было изучено создание собственных прерываний.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЕ КОДЫ ПРОГРАММ

Название файла: lb5.asm

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

```
.186
```

```
STACK SEGMENT STACK
```

```
    DW 1024 DUP(?)
```

```
STACK ENDS
```

```
DATA SEGMENT
```

```
    KEEP_CS DW 0
```

```
    KEEP_IP DW 0
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
    print_cmos proc near
```

```
CMOS      out      70h,al      ; послать AL в индексный порт
```

```
          in       al,71h      ; прочитать данные
```

```
          push     ax
```

```
          shr      al, 4        ; выделить старшие четыре бита
```

```
          add      al,'0'       ; добавить ASCII-код цифры 0
```

```
          int      29h          ; вывести на экран
```

```
          pop      ax
```

```
          and      al,0Fh       ; выделить младшие четыре бита
```

```
          add      al,30h       ; добавить ASCII-код цифры 0
```

```
          int      29h          ; вывести на экран
```

```
          ret
```

```
    print_cmos endp
```

```
inter PROC FAR
```

```
    push     ax
```

```
    push     bx
```

```
    push     cx
```

```

        push    dx
        push    ds

        mov     al,0Bh                ; CMOS 0Bh - управляющий
регистр В
        out     70h,al                ; порт 70h - индекс CMOS
        in      al,71h                ; порт 71h - данные CMOS
        and     al,11111011b          ; обнулить бит 2 (форма чисел -
BCD)
        out     71h,al                ; и записать обратно
        mov     al,4h                 ; CMOS 04h - час
        call    print_cmos
        mov     al,':'                ; двоеточие
        int     29h
        mov     al,2h                 ; CMOS 02h - минута
        call    print_cmos
        mov     al,':'                ; двоеточие
        int     29h
        mov     al,0h                 ; CMOS 00h - секунда
        call    print_cmos
        mov     al, 10
        int     29h
        mov     al, 13
        int     29h

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

inter    ENDP

main    PROC    FAR

```

```

mov ah, 35h
mov al, 23h
int 21h
mov KEEP_IP, bx
mov KEEP_CS, es

push ds
mov dx, offset inter
mov ax, seg inter
mov ds, ax
mov ah, 25h
mov al, 23h
int 21h
pop ds

begin:
    mov ah, 0
    int 16h
    cmp al, 'q'
    je quit
    cmp al, 3
    jnz begin

    int 23h
    jmp begin

quit:

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
```

ПРИЛОЖЕНИЕ В

ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lr5.lst

#Microsoft (R) Macro Assembler Version 5.10 12/1/21

17:45:26

Page

1-1

ASSUME

CS:CODE, DS:DATA, SS:STACK

.186

0000

STACK SEGMENT

STACK

0000 0400 [

DW 1024

DUP(?)

????

]

0800

STACK ENDS

0000

DATA SEGMENT

0000 0000

KEEP_CS DW 0

0002 0000

KEEP_IP DW 0

0004

DATA ENDS

0000

CODE SEGMENT

0000

print_cmos proc near

0000 E6 70

out

70h,al

; поэ

		завить ASCII-
код цифры 0		
0011 CD 29	int	29h
; вый		
		ести на
экран		
0013 C3	ret	
0014		
	print_cmos endp	
0014		inter
PROC FAR		
0014 50	push	ax
0015 53		push
bx		
0016 51		push
cx		
0017 52		push
dx		
0018 1E		push
ds		
0019 B0 0B		mov
al,0Bh		
		; CMOS 0Bh -
управляющий регис		
		тр B
001B E6 70	out	
70h,al		; поэ
		от 70h -

индекс CMOS

001D E4 71

in

al,71h

; поэ

От 71h -

данные CMOS

17:45:26

Page

1-2

```
001F 24 FB                                and
al,11111011b                            ; обЪИ

                                           1/2улить бит 2

(форма чисел - B

                                           CD)

0021 E6 71                                out
71h,al                                    ; и з

                                           аписать

обратно

0023 B0 04                                mov
al,4h                                    ; CMOS

                                           04h - час

0025 E8 0000 R                            call    print_cmos
0028 B0 3A                                mov
al,':'                                  ; двЪИ

                                           3/4еточие

002A CD 29                                int      29h
002C B0 02                                mov
al,2h                                    ; CMOS

                                           02h - минута

002E E8 0000 R                            call    print_cmos
0031 B0 3A                                mov
al,':'                                  ; двЪИ
```

			методиче
0033	CD 29	int	29h
0035	B0 00	mov	
al, 0h			
			00h - секунда
0037	E8 0000 R	call	print_cmos
003A	B0 0A	mov	
al, 10			
003C	CD 29	int	29h
003E	B0 0D	mov	
al, 13			
0040	CD 29	int	29h
0042	58	pop	ax
0043	5B	pop	bx
0044	59	pop	cx
0045	5A	pop	dx
0046	1F	pop	ds
0047	B0 20	mov	al, 20h
0049	E6 20	out	20h, al
004B	CF	iret	
004C			inter
ENDP			
004C			main PROC
FAR			
004C	B4 35	mov	ah,
35h			
004E	B0 23	mov	al,

23h

0050 CD 21 int 21h

0052 89 1E 0002 R mov KEEP_IP,

bx

0056 8C 06 0000 R mov KEEP_CS,

es

005A 1E push ds

005B BA 0014 R mov dx,

offset inter

005E B8 ---- R mov ax, seg

inter

0061 8E D8 mov ds,

ax

0063 B4 25 mov ah,

25h

0065 B0 23 mov al,

23h

0067 CD 21 int 21h

0069 1F pop ds

17:45:26

Page

1-3

006A

begin:

006A B4 00 mov
ah,0

006C CD 16 int
16h

006E 3C 71 cmp
al, 'q'

0070 74 08 je
quit

0072 3C 03 cmp
al,3

0074 75 F4 jnz
begin

0076 CD 23 int
23h

0078 EB F0 jmp
begin

007A

quit:

007A FA cli

007B	1E	push ds
007C	8B 16 0002 R	mov dx,
KEEP_IP		
0080	A1 0000 R	mov ax,
KEEP_CS		
0083	8E D8	mov ds,
ax		
0085	B4 25	mov ah,
25h		
0087	B0 23	mov al,
23h		
0089	CD 21	int 21h
008B	1F	pop ds
008C	FB	sti
008D	B4 4C	mov ah,
4ch		
008F	CD 21	int 21h
0091		main ENDP
0091		CODE ENDS
		END main

17:45:26

Symbols-

1

Segments and Groups:

N a m e	Length	Align	Combine Class
CODE	0091	PARA	NONE
DATA	0004	PARA	NONE
STACK	0800	PARA	STACK

Symbols:

N a m e	Type	Value	Attr
BEGIN	L NEAR	006A	CODE
INTER	F PROC	0014	CODE Length = 0038
KEEP_CS	L WORD	0000	DATA
KEEP_IP	L WORD	0002	DATA
MAIN	F PROC	004C	CODE Length = 0045

PRINT_CMOS N PROC 0000 CODE Length =
0014

QUIT L NEAR 007A CODE

@CPU TEXT 259

@FILENAME TEXT 1b5

@VERSION TEXT 510

109 Source Lines

109 Total Lines

15 Symbols

48034 + 459226 Bytes symbol space free

0 Warning Errors

0 Severe Errors