

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: Представление и обработка символьной информации с
использованием строковых команд

Студент гр. 0384

Кусмарцев А.И.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Научиться создавать обработки прерывания.

Задание.

Код задания 2е.

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 работы.

Вариант 12.

Для задания адреса собственного прерывания с заданным номером в таблицу векторов прерываний используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес нового обработчика.

В самом векторе прерывания будем считывать из порта cmos значения часов, минут, секунд. Затем будет выводить 0, 1 формируя строку вида часы, минуты, секунды в формате BCD

Результаты тестирования приведены ниже в таблице.

Входная строка	Результат	Комментарий
	00010010;00100111;01001000	Верно(запуск осуществился в 12:27)

Разработанный программный код см. в приложении А.

Выводы.

В ходе выполнения работы были изучены способы создания собственных прерываний.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
Astack SEGMENT STACK
    DW 1024    DUP(?)
Astack ENDS
;перед выполнением прерывания нужно сохранить в стек значение
CS::IP, чтобы к ней обратиться в последствии
;a в cs::ip записать адрес программы обработки прерывания
DATA SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0

DATA ENDS

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

;вектор прерывания это программа по обработке прерывания
;ее адрес состоит из 4 байт, в первых двух хранится значение IP,
а во вторых - CS

print_cmos PROC FAR

    out        70h,al                ; послать AL в индексный
порт CMOS
    in         al,71h                ; прочитать данные
    mov cl, 128 ;будем проверять через значение в cx
    print:
    push ax
    and al, cl
    cmp al,0
    je zero

    mov al, '1'
    jmp continue

    zero:
    mov al, 48
    continue:
    int 29h
    pop ax
    shr cl, 1
    cmp cl, 1
    jge print
    jmp end
    end:
    ret
print_cmos endp

MY_INT PROC FAR
```

```

push ax
push bx
push cx
mov al, 08h
out 70h, al
in al, 71h

mov al, 4h

call print_cmos

mov al, ';'
int 29h

mov al, 2h

call print_cmos

mov al, ';'
int 29h

mov al, 0h

call print_cmos
pop cx
pop bx
pop ax

mov al, 20H
out 20H, al
iret

```

MY_INT ENDP

MAIN PROC FAR

```

push ds
sub ax, ax
push ax
mov ax, DATA
mov ds, ax

; записывание текущего вектора прерывания
mov ah, 35H ;функция получения вектора
mov al, 1CH;номер вектора
int 21h
mov KEEP_IP, bx; запоминания смещения
mov KEEP_CS, es; и сегмента вектора прерывания

; установка нового вектора прерывания
push ds
mov dx, OFFSET MY_INT;смещения для процедуры в DX
mov ax, SEG MY_INT; сегмент процедуры
mov ds, ax; помещаем сегмент процедуры в ds
mov ah, 25H; функция установки вектора
mov al, 60H; номер вектора

```

```

        int 21H; меняем прерывание
        pop ds

        int 60H

        CLI
        push ds
        mov dx, KEEP_IP
        mov ax, KEEP_CS
        mov ds, ax
        mov ah, 25H
        mov al, 1CH
        int 21H; восстанавливаем старый вектор прерывания
        pop ds
        sti

MAIN ENDP
CODE ENDS
END MAIN

```

Название файла: lab5.lst

#Microsoft	(R)	Macro	Assembler	Version	5.10
12/23/21 12:44:5					
1-1				Page	

```

0000                                Astack SEGMENT STACK
0000 0400[                            DW 1024      DUP(?)
      ????                            ]

0800                                Astack ENDS
;Ð¿ÐµÑÑ ÐµÐŹ Ð²Ñ Ð¿ÐŸÐ»ÐµÐµÐŹÐµÐŹ
Ð¿Ñ ÐµÑÑÑÑÑ Ð²
ÑÑÐ
µÐ° Ð·Ð°Ð°ÑÑ ÐµÐµÐŹÐµ CS::IP, ÑÑÐŸÐ±Ñ Ð°
ÐµÐµ
Ð¹ ÐŸÐ±Ñ Ð°ÑÐŹÑÑÑÑÑ Ð²
Ð¿
;Ð° Ð² cs::ip Ð·Ð°Ð¿ÐŹÑÐ°ÑÑÑ Ð°ÐŹÑÑ ÐµÑÑ·
ŸÐ³Ñ Ð°ÐµÐµÑÑ ÐŸÐ±Ñ Ð°Ð±ŸÑÑ Ð°ÐŹ
ÐµÐŹÑÑ
DATA SEGMENT
0000                                KEEP_CS DW 0
0000 0000                                KEEP_IP DW 0
0002 0000

```

0004		DATA ENDS
0000		CODE SEGMENT
		ASSUME CS:CODE, DS:DATA, SS:Astack
ĐŁŃ Đ		;Đ²ĐµĐ°Ń-ĐŸŃ ĐŁŃ ĐµŃ Ń Đ²Đ°ĐœĐŽŃ Ń Ń-ĐŸ
ĐŁŃ ĐµŃ Ń		ŸĐ³Ń Đ°ĐœĐœĐ° ĐŁĐŸ ĐŸĐ±Ń Đ°Đ±ĐŸŃ Đ°Đµ
Đ±Đ°Đ¹Ń		Đ²Đ°ĐœĐŽŃ
Đ·Đ		;ĐµĐµ Đ°ĐŽŃ ĐµŃ· Ń·ĐŸŃ·Ń-ĐŸĐŽŃ- ĐŽĐ· 4
		, Đ² ĐŁĐµŃ Đ²Ń Ń ĐŽĐ²Ń Ń Ń Ń Đ°ĐœĐŽŃ-Ń·Ń
		œĐ°Ń ĐµĐœĐŽĐµ IP, Đ° Đ²ĐŸ Đ²Ń-ĐŸŃ Ń Ń - CS
0000		print_cmos PROC FAR
0000 E6 70		out 70h, al
; ĐŁĐŸŃ		·Đ»Đ°Ń-Ń AL Đ² ĐŽĐœĐŽĐµĐ°Ń-ĐœŃ Đ¹ ĐŁĐŸŃ Ń-
CMO		S
0002 E4 71		in al, 71h
; ĐŁŃ Đ		ŸŃ ĐŽŃ-Đ°Ń-Ń ĐŽĐ°ĐœĐœŃ Đµ
0004 B1 80		mov cl, 128 ;Đ±Ń ĐŽĐµĐœ
ĐŁŃ ĐŸĐ²ĐµŃ Ń Ń-Ń		Ń ĐµŃ ĐµĐ· Đ·ĐœĐ°Ń ĐµĐœĐŽĐµ Đ² cx
0006		print:
0006 50		push ax
0007 22 C1		and al, cl
0009 3C 00		cmp al, 0
000B 74 05		je zero
000D B0 31		mov al, '1'
000F EB 03 90		jmp continue
0012		zero:
0012 B0 30		mov al, 48
0014		continue:
0014 CD 29		int 29h
#Microsoft (R)	Macro	Assembler Version 5.10
12/23/21 12:44:5		
1-2		Page
0016 58		pop ax
0017 D0 E9		shr cl, 1
0019 80 F9 01		cmp cl, 1
001C 7D E8		jge print
001E EB 01 90		jmp end
0021		end:

lab5.asm(43): warning A4016: Reserved word used as symbol: END

```
0021 CB                ret
0022                print_cmos endp

0022                MY_INT PROC FAR
0022 50                push ax
0023 53                push bx
0024 51                push cx
0025 B0 08            mov al, 08h
0027 E6 70            out 70h, al
0029 E4 71            in al, 71h

002B B0 04            mov al, 4h

002D 9A 0000 ---- R   call print_cmos

0032 B0 3B            mov al, ';'
0034 CD 29            int 29h

0036 B0 02            mov al, 2h

0038 9A 0000 ---- R   call print_cmos

003D B0 3B            mov al, ';'
003F CD 29            int 29h

0041 B0 00            mov al, 0h

0043 9A 0000 ---- R   call print_cmos
0048 59                pop cx
0049 5B                pop bx
004A 58                pop ax

004B B0 20            mov al, 20H
004D E6 20            out 20H, al
004F CF                iret

0050                MY_INT ENDP

0050                MAIN PROC FAR
0050 1E                push ds
0051 2B C0            sub ax, ax
0053 50                push ax
0054 B8 ---- R        mov ax, DATA
0057 8E D8            mov ds, ax
```

Ń-ĐμĐ°Ń Ń ĐμĐ³ĐŸ Đ ; Đ·Đ°ĐŁĐŽŃ·Ń Đ²Đ°ĐæĐŽĐμ
²ĐμĐ°Ń-ĐŸŃ Đ° ĐŁŃ ĐμŃ Ń Đ²Đ°ĐæĐŽŃ

```

0059 B4 35 mov ah, 35H ; Ĩ__Ĩ ԹԹԹԹ Ĩ ԶՃĨ
ԶԶԶԶ»Ĩ Ĩ ԹԹԹԹ

005B B0 1C mov al, 1CH; ԹԹԶԶԹԹԹ
ԶԶԹԹԹԹ Ĩ ԶԶԶԹ Թ
005D CD 21 int 21h
005F 89 1E 0002 R mov KEEP_IP, bx;
Թ·ԹԹԶԶԶԶԶԶԶԶԶԶԶ Ĩ Ĩ·Զ
ԹԹԹԹ ԹԹԶԶԶՃ
0063 8C 06 0000 R mov KEEP_CS, es; ԶՃ
Ĩ·ԹԹԶԶԶԶԶԶԶԶԶԶ ԶԶԹԹԹ
Ĩ ԶԶԶԹ Թ ԶԶԶԹ ԹԹ ԶԶԶԶԶԶԶԶ
; Ĩ Ĩ·Ĩ-ԹԹԶԶԶԶԶԶԶ ԹԶԶԶԶԶԶԶԶԶԶ
ԶԶԹԹԹԹ Թ
ԶԶԶԶ 1E push ds
0068 BA 0022 R mov dx, OFFSET MY_INT; Ĩ·ԶԶԹԹԹ ԹԹԶԶԶԶԶ
ԶՃԶ»
006B B8 ---- R Ĩ ԶԶԶԹ ԶԶԶԹ ԹԹԶԶԶ Ĩ Ĩ ԶԶ DX
ԶԶԶ Ĩ ԶԶԶ mov ax, SEG MY_INT; Ĩ·ԹԹԶԶԶԶԶԶԶԶԶ
ԹԹԶԶԶ Ĩ Ĩ
006E 8E D8 mov ds, ax; ԶԶԶԶԶԶԶԶԶ ԹԹԹԶԶԶ
Ĩ·ԹԹԶԶԶԶԶԶԶԶԶԶ
ԶԶԶ Ĩ ԶԶԶ ԹԹԶԶԶ Ĩ Ĩ ԶԶ ds
0070 B4 25 mov ah, 25H; Ĩ__Ĩ ԹԹԹԹ Ĩ ԶՃĨ
Ĩ Ĩ·Ĩ-ԹԹԶԶԶԶԶԶ
ԹՃ ԶԶԶԹԹԹ Ĩ ԶԶԶԹ Թ
0072 B0 60 mov al, 60H; ԹԹԶԶԶԶԶԶԶԶԶ
ԶԶԹԹԹԹ Ĩ ԶԶԶԹ Թ
0074 CD 21 int 21H; ԶԶԶԶԶԶԶ ԹԹԶԶԶ
ԶԶԶ ԹԹԶ Ĩ ԶԶԶԶԶԶԶԶԶԶԶ
0076 1F pop ds
0077 CD 60 int 60H
0079 FA CLI
007A 1E push ds
007B 8B 16 0002 R mov dx, KEEP_IP
007F A1 0000 R mov ax, KEEP_CS
0082 8E D8 mov ds, ax
0084 B4 25 mov ah, 25H
0086 B0 1C mov al, 1CH
0088 CD 21 int 21H;
ԶԶԶԶ·Ĩ·Ĩ-ԹԹԶԶԶԶԶԶԶԶԶԶԶԶԶ ԶԶԶ ԶԶԶ
ԶԶԶ Ĩ ԶԶԶ ԶԶԶԶԶԶԶ ԶԶԶ ԶԶԶ Ĩ ԶԶԶԶԶԶԶԶԶԶԶ
008A 1F pop ds
008B FB sti

```



```

008C          MAIN ENDP
008C          CODE ENDS
              END MAIN
#Microsoft    (R)      Macro      Assembler      Version      5.10
12/23/21 12:44:5
ols-1
Symb

```

Segments and Groups:

Class	N a m e	Length	Align	Combine
	ASTACK	0800	PARA	STACK
	CODE	008C	PARA	NONE
	DATA	0004	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
	CONTINUE	L NEAR	0014	CODE
	END	L NEAR	0021	CODE
	KEEP_CS	L WORD	0000	DATA
	KEEP_IP	L WORD	0002	DATA
= 003C	MAIN	F PROC	0050	CODE Length
= 002E	MY_INT	F PROC	0022	CODE Length
	PRINT	L NEAR	0006	CODE
= 0022	PRINT_CMOS	F PROC	0000	CODE Length
	ZERO	L NEAR	0012	CODE
	@CPU	TEXT	0101h	
	@FILENAME	TEXT	lab5	
	@VERSION	TEXT	510	

```

123 Source  Lines
123 Total   Lines
17 Symbols

```

48002 + 457208 Bytes symbol space free

```

1 Warning Errors
0 Severe  Errors

```