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

Студент гр. 1381 _____ Хомутинников Н.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Написать собственное прерывание на Языке Ассемблера.

Задание.

Вариант 26:

4е — по прерыванию от клавиатуры 16h выполнить чтение и ввод на экран отсчета часов реального времени из памяти CMOS (в формате BCD)

Выполнение работы.

Была написана процедура SUBR_INT для реализации прерывания, в котором перед началом обработки самого прерывания сохраняются изначальные регистры. Затем было использовано прерывание 1Ah с серивисом AH = 02h, позволяющим читать время из постоянных CMOS часов реального времени в формате BCD. После этого поочередно происходит вывод данных из регистров CX и DH, преобразованных из BCD формата в ASCII символы соответствующим им числам.

Преобразование происходит с помощью разделения искомого числа на разряды по регистрам AH и AL и дальнейшего их обращения в ASCII символ через сложение со значением 0 в таблице.

Вывод на экран происходит с использованием прерывания 21h.

В функции Main сохраняются исходные значения нынешнего вектора прерывания 60h(его номер и вектор) с помощью функции 25h/INT 21h. Далее вызывается само измененное прерывание и, по завершению его работы, восстанавливается его исходное значение.

Исходный код программы см. в Приложении А

Таблица 1 – Результаты работы программы lab5

Входные данные	Выходные данные	Комментарий
19:36 q	19:36:22	Верно

19:38 q	19:38:34	Верно
19:41 q	19:41:09	Верно
19:43 q	19:43:30	Верно
19:44 q	19:44:29	Верно

Выводы.

В ходе выполнения лабораторной работы были изучены принципы создания собственных прерываний на языке Ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
INTERRUPTION EQU 16H

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

STACK SEGMENT STACK
    DW 1024 DUP(?)
STACK ENDS

DATA SEGMENT
    KEEP_CS DW 0
    KEEP_IP DW 0
DATA ENDS

CODE SEGMENT

SUBR_INT PROC FAR

    JMP start

    INIT_SS DW 0000h
    INIT_SP DW 0000h
    INT_STACK DB 40 DUP(?)

    read_CMOS PROC
          PUSH DX
                ;hours
          MOV AL, CH       ;in CX=HHMM,
          CALL print_bcd
          CALL colon
                ;minutes
          MOV AL, CL
          CALL print_bcd
          CALL colon
                ;seconds
          MOV AL, DH       ;in DH=SS
          CALL print_bcd
          POP DX
          RET
    read_CMOS ENDP

    colon PROC
          MOV DL, ':'
          MOV AH, 02h
          INT 21H
          RET
    colon ENDP

    print_bcd PROC
          PUSH DX
          PUSH CX
```

```
          MOV CL, 4
          MOV AH, AL
          AND AL, 00001111b
          SHR AH, CL
          ADD AL, '0'
          ADD AH, '0'
          MOV DL, AH
          MOV DH, AL
          MOV AH, 02h
          INT 21h
          MOV DL, DH
          INT 21h
          POP CX
          POP DX
          RET
    print_bcd ENDP

start:
;Сохранение в память текущего состояния регистров

    MOV INIT_SP, SP
    MOV INIT_SS, ss
    mov sp, SEG INT_STACK
    mov ss, sp
    PUSH AX
    PUSH CX
    PUSH DX

;процесс прерывания

    MOV AH, 02H
    INT 1Ah
    CALL read_CMOS
    POP DX
    POP CX
    POP AX
    MOV ss, INIT_SS
    MOV SP, INIT_SP
    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

;текущее состояние вектора

    MOV AH, 35H
    MOV AL, 60H
    INT 21H
    MOV KEEP_IP, BX
```

```
    MOV KEEP_CS, ES

;новый вектор смещения

    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

;считывание символа с клавиатуры (в данном случае это q)
    readkey:
         MOV AH, 0
         INT INTERRUPTION
         CMP AH, 16      ;код считывания символа q
         JNE readkey
         INT 60H

;восстановление исходного вектора прерывания
    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
    MOV AH, 4CH
    INT 21H
Main ENDP
CODE ENDS
END Main
```

Название файла: LAB5.LST

```
 = 0016                     INTERRUPTION EQU 16H

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

 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                   SUBR_INT PROC FAR

 0000  EB 6A 90                    JMP start

 0003  0000                        INIT_SS DW 0000h
 0005  0000                        INIT_SP DW 0000h
 0007  0028[                       INT_STACK DB 40 DUP(?)
        ??
                ]


 002F                         read_CMOS PROC
 002F  52                           PUSH DX
                                         ;hours
 0030  8A C5                             MOV AL, CH      ;in CX=
                        HHMM,
 0032  E8 004E R                    CALL print_bcd
 0035  E8 0047 R                    CALL colon
                                         ;minutes
 0038  8A C1                             MOV AL, CL
 003A  E8 004E R                    CALL print_bcd
 003D  E8 0047 R                    CALL colon
                                         ;seconds
 0040  8A C6                             MOV AL, DH      ;in DH=
                        SS
 0042  E8 004E R                    CALL print_bcd
 0045  5A                           POP DX
 0046  C3                           RET
 0047                         read_CMOS ENDP

 0047                         colon PROC
 0047  B2 3A                             MOV DL, ':'
 0049  B4 02                             MOV AH, 02h
 004B  CD 21                             INT 21H
 004D  C3                           RET
 004E                         colon ENDP
```

```
004E                    print_bcd PROC
004E  52                      PUSH DX
004F  51                      PUSH CX
0050  B1 04                        MOV CL, 4
0052  8A E0                      MOV AH, AL
0054  24 0F                      AND AL, 00001111b
0056  D2 EC                      SHR AH, CL
0058  04 30                      ADD AL, '0'
005A  80 C4 30                    ADD AH, '0'
005D  8A D4                      MOV DL, AH
005F  8A F0                      MOV DH, AL
0061  B4 02                      MOV AH, 02h
0063  CD 21                      INT 21h
0065  8A D6                      MOV DL, DH
0067  CD 21                      INT 21h
0069  59                      POP CX
006A  5A                      POP DX
006B  C3                      RET
006C                    print_bcd ENDP

006C                start:
                    ;PЎPsC…CЂP°PSPµPSPёPµ PI PїP°PjCЏC,CЊ C,PµPєCґC
                    ‰PµPiPs CЃPsCЃC,PsCЏPSPёCЏ CЂPµPiPёCЃC,CЂPsPI

006C  2E: 89 26 0005 R              MOV INIT_SP, SP
0071  2E: 8C 16 0003 R              MOV INIT_SS, ss
0076  BC ---- R           mov sp, SEG INT_STACK
0079  8E D4                    mov ss, sp
007B  50                  PUSH AX
007C  51                  PUSH CX
007D  52                  PUSH DX

                    ;PїCЂPsC†PµCЃCЃ PїCЂPµCЂC‹ PIP°PSPёCЏ

007E  B4 02                      MOV AH, 02H
0080  CD 1A                      INT 1Ah
0082  E8 002F R           CALL read_CMOS
0085  5A                  POP DX
0086  59                  POP CX
0087  58                  POP AX
0088  2E: 8E 16 0003 R              MOV ss, INIT_SS
008D  2E: 8B 26 0005 R              MOV SP, INIT_SP
0092  B0 20                      MOV AL, 20H
0094  E6 20                      OUT 20H, AL
0096  CF                  IRET

0097                SUBR_INT ENDP


0097                Main  PROC FAR
0097  1E                      PUSH DS
0098  2B C0                      SUB AX, AX
009A  50                  PUSH AX
```

```
009B  B8 ---- R           MOV AX, DATA
009E  8E D8                     MOV DS, AX

                    ;C,PµPєCґC‰PµPµ CЃPsCЃC,PsCЏPSPёPµ PIPµPєC,PsCЋ
                    P°

00A0  B4 35                     MOV AH, 35H
00A2  B0 60                     MOV AL, 60H
00A4  CD 21                     INT 21H
00A6  89 1E 0002 R              MOV KEEP_IP, BX
00AA  8C 06 0000 R              MOV KEEP_CS, ES

                    ;PSPsPIC‹PЊ PIPµPєC,PsCЋ CЃPjPµC‰PµPSPёCЏ

00AE  1E                  PUSH DS
00AF  BA 0000 R           MOV DX, OFFSET SUBR_INT
00B2  B8 ---- R           MOV AX, SEG SUBR_INT
00B5  8E D8                     MOV DS, AX
00B7  B4 25                     MOV AH, 25H

00B9  B0 60                     MOV AL, 60H
00BB  CD 21                     INT 21H
00BD  1F                  POP DS

                    ;CЃC‡PёC,C‹PIP°PSPёPµ CЃPёPjPIPsP»P° CЃ PєP»P°P
                    IPёP°C,CґCЋC‹ (PI PrP°PSPSPsPj CЃP»CґC‡P°Pµ CЌC
                    ,Ps q)
00BE                      readkey:
00BE  B4 00                         MOV AH, 0
00C0  CD 16                         INT INTERRUPTION

00C2  80 FC 10                      CMP AH, 16      ;PєPsPr
                     CЃC‡PёC,C‹PIP°PSPёCЏ CЃPёPjPIPsP»P° q
00C5  75 F7                         JNE readkey
00C7  CD 60                         INT 60H

                    ;PIPsCЃCЃC,P°PSPsPIP»PµPSPёPµ PёCЃC…PsPrPSPsPiP
                    s PIPµPєC,PsCЋP° PїCЋPµCЋC‹PIP°PSPёCЏ
00C9  FA                  CLI
00CA  1E                  PUSH DS
00CB  8B 16 0002 R              MOV DX, KEEP_IP
00CF  A1 0000 R           MOV AX, KEEP_CS
00D2  8E D8                     MOV DS, AX
00D4  B4 25                     MOV AH, 25H
00D6  B0 60                     MOV AL, 60H
00D8  CD 21                     INT 21H
00DA  1F                  POP DS
00DB  FB                  STI
00DC  B4 4C                     MOV AH, 4CH
00DE  CD 21                     INT 21H
00E0                Main ENDP
00E0                CODE ENDS
                    END Main
```

Segments and Groups:

N a m e	Length	Align	Combine Class
CODE	00E0	PARA	NONE
DATA	0004	PARA	NONE
STACK	0800	PARA	STACK

Symbols:

N a m e	Type	Value	Attr	
COLON	N PROC	0047	CODE	Length = 0007
INIT_SP	L WORD	0005	CODE	
INIT_SS	L WORD	0003	CODE	
INTERRUPTION	NUMBER	0016		
INT_STACK	L BYTE	0007	CODE	Length = 0028
KEEP_CS	L WORD	0000	DATA	
KEEP_IP	L WORD	0002	DATA	
MAIN	F PROC	0097	CODE	Length = 0049
PRINT_BCD	N PROC	004E	CODE	Length = 001E
READKEY	L NEAR	00BE	CODE	
READ_CMOS	N PROC	002F	CODE	Length = 0018
START	L NEAR	006C	CODE	
SUBR_INT	F PROC	0000	CODE	Length = 0097
@CPU	TEXT	0101h		
@FILENAME	TEXT	lab5		
@VERSION	TEXT	510		

145 Source Lines
145 Total Lines
21 Symbols

48020 + 459240 Bytes symbol space free

0 Warning Errors
0 Severe Errors