

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: Разработка собственного прерывания.

Студент гр. 9383

Ноздрин В.Я.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Познакомиться с прерываниями, изучить их работу, разработать собственное прерывание.

Задание

Вариант 4А:

4: Номер и назначение заменяемого вектора прерывания - 08h - прерывание от системного таймера - генерируется автоматически операционной системой 18 раз в сек.

А: Действие, реализуемое программой обработки прерывания — печать сообщения на экране.

Ход работы.

В ходе работы была написана программа на языке Ассемблер, которая сохраняет старый вектор прерывания, устанавливает новый, вызывает его обработку и восстанавливает старый вектор прерывания.

В программе реализована отдельная процедура SUBR_INT, которая является программой обработки прерывания. В ней по заданию выводится на экран строка с текстом «This is my INT msg». Также используются следующие инструкции:

- OUT - выводит данные из регистра AL или AX (ИСТОЧНИК) в порт ввода-вывода.
- IRET - возврат обработки прерывания.

В процедуре MAIN происходит сохранение старого, установка нового и восстановление старого векторов прерывания. Используются следующие функции и инструкции:

- 35h - возвращает текущее значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX.
- 25h - помещает заданные адреса сегмента и смещения обработчика в вектор прерывания с заданным номером.

- CLI - сбрасывает флаг прерывания IF. Выполнение этой команды отключает аппаратные прерывания.
- STI - устанавливает флаг разрешения прерываний IF в регистре флагов, разрешая все аппаратные прерывания.

Исходный код см. в приложении А.

Выводы.

Было произведено знакомство с прерываниями, была изучена их работу, было разработано собственное прерывание.

ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММЫ

Текст файла main.asm

```
AStack SEGMENT STACK
    DW 1024 DUP(?)    ; 1 килобайт
AStack ENDS

DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения вектора прерывания
    MSG      DB "This is my INT msg",10,13,'$'
DATA ENDS

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

SUBR_INT PROC FAR
    jmp ISTART
    KEEP_SP DW 0
    KEEP_SS DW 0
    KEEP_AX DW 0
    ISTACK  DW 16 DUP(?)
ISTART:
    MOV KEEP_SP, SP
    MOV KEEP_AX, AX
    MOV AX, SS
    MOV KEEP_SS, AX
    MOV AX, KEEP_AX
    MOV SP, OFFSET ISTART
    MOV AX, SEG ISTACK
    MOV SS, AX

    PUSH AX                ; сохранение изменяемых регистров
    PUSH DX
        ; <действия по обработке прерывания>
    MOV DX, OFFSET MSG
    MOV AH, 09H
    INT 21H
    POP DX
    POP AX
```

```

MOV KEEP_AX, 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

Main PROC FAR

```

MOV AX, DATA
MOV DS, AX

```

```

MOV AH, 35H           ; ф-ия получения вектора
MOV AL, 1CH           ; номер прерывания
INT 21H
MOV KEEP_IP, BX       ; запоминание смещения
MOV KEEP_CS, ES       ; и сегмента вектора прерывания

```

```

PUSH DS               ; сохранение DS
MOV DX, OFFSET SUBR_INT ; смещение для процедуры в DX
MOV AX, SEG SUBR_INT   ; сегмент процедуры через AX
MOV DS, AX             ; в DS
MOV AH, 25H           ; функция установки вектора
MOV AL, 1CH           ; номер вектора
MOV AL, 60H           ; номер вектора
MOV AL, 23H           ; номер вектора
MOV AL, 08H           ; номер вектора
INT 21H               ; меняем прерывание
POP DS                ; восстановление DS

```

; Далее может выполняться вызов нового обработчика прерывания.

label1:

```

MOV AH, 01H
INT 21H
cmp AL, 21
jne label1

```

; В конце программы восстанавливается старый вектор прерывания

```

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

MOV AH, 4CH
INT 21H
Main ENDP
CODE ENDS
END Main

```

Текст файла main.lst

Microsoft (R) Macro Assembler Version 5.10

12/24/20 00:10:4

Page 1-1

```

0000          AStack SEGMENT STACK
0000 0400[          DW 1024 DUP(?)    ; 1 килобайт      ???? ]
0800          AStack ENDS

0000          DATA SEGMENT
0000 0000          KEEP_CS DW 0 ; для хранения сегмента
0002 0000          KEEP_IP DW 0 ; и смещения вектора прерывания
0004 54 68 69 73 20 69      MSG      DB "This is my INT msg",10,13,'$'
          73 20 6D 79 20 49
          4E 54 20 6D 73 67
          0A 0D 24
0019          DATA ENDS

0000          CODE SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

0000          SUBR_INT PROC FAR
0000 EB 27 90          jmp ISTART
0003 0000          KEEP_SP DW 0
0005 0000          KEEP_SS DW 0
0007 0000          KEEP_AX DW 0
0009 0010[          ISTACK DW 16 DUP(?)    ???? ]

0029          ISTART:
0029 2E: 89 26 0003 R          MOV KEEP_SP, SP
002E 2E: A3 0007 R          MOV KEEP_AX, AX
0032 8C D0          MOV AX, SS

```

```

0034 2E: A3 0005 R      MOV KEEP_SS, AX
0038 2E: A1 0007 R      MOV AX, KEEP_AX
003C BC 0029 R      MOV SP, OFFSET ISTART
003F B8 ---- R      MOV AX, SEG ISTACK
0042 8E D0      MOV SS, AX

0044 50      PUSH AX      ; сохранение изменяемых регистров
0045 52      PUSH DX
      ;<действия по обработке прерывания>
0046 BA 0004 R      MOV DX, OFFSET MSG
0049 B4 09      MOV AH, 09H
004B CD 21      INT 21H
004D 5A      POP DX
004E 58      POP AX

```

Microsoft (R) Macro Assembler Version 5.10

12/24/20 00:10:4

Page 1-2

```

004F 2E: A3 0007 R      MOV KEEP_AX, AX
0053 2E: 8B 26 0003 R      MOV SP, KEEP_SP
0058 2E: A1 0005 R      MOV AX, KEEP_SS
005C 8E D0      MOV SS, AX
005E 2E: A1 0007 R      MOV AX, KEEP_AX

0062 B0 20      MOV AL, 20H
0064 E6 20      OUT 20H, AL
0066 CF      IRET
0067      SUBR_INT ENDP

```

```

0067      Main PROC FAR
0067 B8 ---- R      MOV AX, DATA
006A 8E D8      MOV DS, AX

006C B4 35      MOV AH, 35H      ; ф-ия получения вектора
006E B0 1C      MOV AL, 1CH      ; номер прерывания
0070 CD 21      INT 21H
0072 89 1E 0002 R      MOV KEEP_IP, BX      ; запоминание смещения
0076 8C 06 0000 R      MOV KEEP_CS, ES      ; и сегмента вектора
прерывания
007A 1E      PUSH DS      ; сохранение DS
007B BA 0000 R      MOV DX, OFFSET SUBR_INT      ; смещение для процедуры в DX
007E B8 ---- R      MOV AX, SEG SUBR_INT      ; сегмент процедуры через AX
0081 8E D8      MOV DS, AX      ; в DS
0083 B4 25      MOV AH, 25H      ; функция установки вектора
0085 B0 1C      MOV AL, 1CH      ; номер вектора
0087 B0 60      MOV AL, 60H      ; номер вектора

```

```

0089  B0 23          MOV AL, 23H          ; номер вектора
008B  B0 08          MOV AL, 08H          ; номер вектора
008D  CD 21          INT 21H             ; меняем прерывание
008F  1F            POP DS               ; восстановление DS

; Далее может выполняться вызов нового обработчика прерывания.
0090          label1:
0090  B4 01          MOV AH, 01H
0092  CD 21          INT 21H
Microsoft (R) Macro Assembler Version 5.10      12/24/20 00:10:4
                                                Page      1-3

0094  3C 15          cmp AL, 21
0096  75 F8          jne label1

; В конце программы восстанавливается старый вектор прерывания
0098  FA            CLI
0099  1E            PUSH DS
009A  8B 16 0002 R    MOV DX, KEEP_IP
009E  A1 0000 R      MOV AX, KEEP_CS
00A1  8E D8          MOV DS, AX
00A3  B4 25          MOV AH, 25H
00A5  B0 1C          MOV AL, 1CH
00A7  CD 21          INT 21H             ; восстанавливаем старый вектор
прерывания
00A9  1F            POP DS
00AA  FB            STI

00AB  B4 4C          MOV AH, 4CH
00AD  CD 21          INT 21H
00AF          Main ENDP
00AF          CODE ENDS
          END Main
Microsoft (R) Macro Assembler Version 5.10      12/24/20 00:10:4
                                                Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0800	PARA	STACK
CODE	00AF	PARA	NONE
DATA	0019	PARA	NONE

Symbols:

<i>N a m e</i>	<i>Type</i>	<i>Value</i>	<i>Attr</i>	
ISTACK	L WORD	0009	CODE	Length = 0010
ISTART	L NEAR	0029	CODE	
KEEP_AX	L WORD	0007	CODE	
KEEP_CS	L WORD	0000	DATA	
KEEP_IP	L WORD	0002	DATA	
KEEP_SP	L WORD	0003	CODE	
KEEP_SS	L WORD	0005	CODE	
LABEL1	L NEAR	0090	CODE	
MAIN	F PROC	0067	CODE	Length = 0048
MSG	L BYTE	0004	DATA	
SUBR_INT	F PROC	0000	CODE	Length = 0067
@CPU	TEXT	0101h		
@FILENAME	TEXT	MAIN		
@VERSION	TEXT	510		

100 Source Lines

100 Total Lines

19 Symbols

48000 + 459257 Bytes symbol space free

0 Warning Errors

0 Severe Errors