

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. 0383

Преподаватель

Коротков А.В.

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Вариант 3b:

Создать собственное прерывание с номером 16h - прерывание, генерируемое при нажатии клавиш Control + C и выполняющее произведение звукового сигнала с заданной высотой звука.

Выполнение работы.

Для хранения сегмента и смещения прерывания создаются переменные KEEP_CS и KEEP_IP. В сегменте кода создан обработчик прерывания my_int, процедуры обработки прерывания. Также задается собственный стек для обработчика прерывания. Процедура sound позволяет произвести звук с заданной частотой.

Для произведения звука используется функция MS-DOS.

Прерывание меняется следующим образом: с помощью функции 35h прерывания 21h получаем необходимый вектор прерывания 23h. Затем записываем этот вектор в переменные KEEP_CS и KEEP_IP. Далее с помощью функции 25h прерывания 21h восстанавливаем предыдущие значения.

Тексты исходного файла программы см. в приложении А.

Тексты диагностических сообщений см. в приложении Б.

Выводы.

В ходе выполнения данной лабораторной работы была изучена работа с прерываниями на языке Ассемблер.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММЫ

Название файла: **lb5.asm**

```
AStack  SEGMENT STACK
```

```
        DB 1024 DUP(?)
```

```
AStack  ENDS
```

```
DATA     SEGMENT
```

```
        KEEP_CS DW 0      ;хранение сегмента
```

```
        KEEP_IP DW 0      ;хранение смещения вектора прерывания
```

```
DATA     ENDS
```

```
CODE     SEGMENT
```

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

```
my_int PROC FAR
```

```
        jmp start
```

```
        KEEP_SS DW 0
```

```
        KEEP_SP DW 0
```

```
        KEEP_AX DW 0
```

```
        IStack DW 30 DUP(?)
```

```
sound proc
```

```
        ;mov AX, 1000
```

```
MOV CX, 1000
```

```
MOV AL, 10110110b ; 0B6H
```

```
OUT 43H, AL ; Код для установления канала 2 таймера-счетчика на работу в
качестве делителя частоты см. методу
```

```
MOV AX, CX ; Заносим в AX высоту звука
```

```
OUT 42H, AL
```

```
MOV AL, AH
```

```
OUT 42H, AL ; Заносим поочередно 2 байта в порт 42h(регистр канала 2)
```

```
IN AL, 61H ; генерация звука путём сдвига диффузора туда-обратно
```

```
MOV AH, AL
```

```

    OR AL, 3
    OUT 61H, AL
    SUB CX, CX
    WHILE_SOUND:
    NOP
    LOOP WHILE_SOUND ; Цикл, пока динамик работает
    MOV AL, AH
    OUT 61H, AL

    ;int 21H
    ret
sound endp

```

```

start:
    MOV KEEP_SP, SP
    MOV KEEP_AX, AX
    MOV AX, SS
    MOV KEEP_SS, AX
    MOV AX, KEEP_AX
    MOV SP, OFFSET start
    MOV SS, AX

    push AX      ; сохранение изменяемых регистров
    push DX

;-----

    int 21h ; вызов прерывания ms-dos
    call sound

;-----

    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
```

```
my_int ENDP
```

```
MAIN PROC FAR
```

```
MOV AX, DATA
MOV DS, AX ; сохраняем вектор прерывания
MOV AH, 35H ; функция получения вектора
MOV AL, 23H ; номер нужного вектора
INT 21H
MOV KEEP_IP, BX ; запоминание смещения
MOV KEEP_CS, ES ; и сегмента вектора прерывания
```

```
PUSH DS
MOV DX, OFFSET my_int ; смещение для процедуры в DX
MOV AX, SEG my_int ; сегмент процедуры
MOV DS, AX; помещаем в DS
MOV AH, 25H; функция установки вектора
MOV AL, 23H; номер вектора
INT 21H; меняем прерывание
POP DS
```

```
ctrl_c:
```

```
mov ah, 0
int 16h
cmp al, 3 ;код символа после нажатия
jne ctrl_c
```

```
INT 23H
```

```

CLI
PUSH DS
MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 23H
INT 21H
; восстанавливаем старый вектор прерывания
POP DS
STI

MOV AH, 4CH
INT 21H

MAIN ENDP
CODE ENDS
END MAIN

```

ПРИЛОЖЕНИЕ Б

ТЕКСТЫ ДИАГНОСТИЧЕСКИХ ФАЙЛОВ ПРОГРАММЫ

Название файла: **lb5.lst**

Microsoft (R) Macro Assembler Version 5.10

12/26/21 17:01:2

Page 1-1

```

0000          AStack  SEGMENT STACK
0000 0400[          DB 1024 DUP(?)

```

??

]

0400 AStack ENDS

0000 DATA SEGMENT

0000 0000 KEEP_CS DW 0 ;хранение сегм
♦нта

0002 0000 KEEP_IP DW 0 ;хранение смещ
♦ния вектора прерывания

0004 DATA ENDS

0000 CODE SEGMENT

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

0000 my_int PROC FAR

0000 EB 64 90 jmp start

0003 0000 KEEP_SS DW 0

0005 0000 KEEP_SP DW 0

0007 0000 KEEP_AX DW 0

0009 001E[IStack DW 30 DUP(?)

 ????

]

0045 time proc

 ;mov AX, 1000

0045 B9 03E8 MOV CX, 1000

0048 B0 B6 MOV AL, 10110110b ; 0B6H

004A E6 43 OUT 43H, AL ; Код для устан♦

♦вления канала 2 таймера-с♦

♦етчика на работу в качест

ве делителя частоты см. ме

тоду

```

004C  8B C1                      MOV AX, CX ; Заносим в AX вы-
                                         -оту звука
004E  E6 42                      OUT 42H, AL
0050  8A C4                      MOV AL, AH
0052  E6 42                      OUT 42H, AL ; Заносим пооче-
                                         редно 2 байта в порт 42h(рег-
                                         -стр канала 2)
0054  E4 61                      IN AL, 61H ; генерация зву-
                                         -а путём сдвига диффузора
                                         туда-обратно
0056  8A E0                      MOV AH, AL
0058  0C 03                      OR AL, 3
005A  E6 61                      OUT 61H, AL
005C  2B C9                      SUB CX, CX
005E                                WHILE_TIME:
005E  90                          NOP
005F  E2 FD                      LOOP WHILE_TIME ; Цикл, пока д-
                                         -инамик работает

```

Microsoft (R) Macro Assembler Version 5.10

12/26/21 17:01:2

Page 1-2

```

0061  8A C4                      MOV AL, AH
0063  E6 61                      OUT 61H, AL

                                         ;int 21H
0065  C3                          ret
0066                                time endp

0066                                start:
0066  2E: 89 26 0005 R              MOV KEEP_SP, SP
006B  2E: A3 0007 R              MOV KEEP_AX, AX
006F  8C D0                      MOV AX, SS
0071  2E: A3 0003 R              MOV KEEP_SS, AX

```



```

0075  2E: A1 0007 R      MOV AX, KEEP_AX
0079  BC 0066 R      MOV SP, OFFSET start
007C  8E D0      MOV SS, AX

007E  50      push AX      ; сохранение изме◆
                           ◆яемых регистров
007F  52      push DX
                           ;-----
0080  CD 21      int 21h ; вызов прерывания m
                           s-dos
0082  E8 0045 R      call time
                           ;-----

0085  5A      pop DX
0086  58      pop AX
0087  2E: A3 0007 R      MOV KEEP_AX, AX
008B  2E: 8B 26 0005 R      MOV SP, KEEP_SP
0090  2E: A1 0003 R      MOV AX, KEEP_SS
0094  8E D0      MOV SS, AX
0096  2E: A1 0007 R      MOV AX, KEEP_AX

009A  B0 20      mov al,20h
009C  E6 20      out 20h,al
009E  CF      iret
009F      my_int ENDP

009F      MAIN PROC FAR
009F  B8 ---- R      MOV AX, DATA
00A2  8E D8      MOV DS, AX ; сохраняем векто
                           р прерывания
00A4  B4 35      MOV AH, 35H ; функция получен
                           ия вектора
00A6  B0 23      MOV AL, 23H ; номер нужного
                           вектора

```

```

00A8  CD 21                      INT 21H
00AA  89 1E 0002 R              MOV KEEP_IP, BX ; запоминани◆
                                   ◆ смещения
00AE  8C 06 0000 R              MOV KEEP_CS, ES ; и сегмента ◆
                                   ◆ектора прерывания

00B2  1E                      PUSH DS
00B3  BA 0000 R              MOV DX, OFFSET my_int ; смещение д
Microsoft (R) Macro Assembler Version 5.10                      12/26/21 17:01:2
                                                                    Page      1-3

                                   ля процедуры в DX
00B6  B8 ---- R              MOV AX, SEG my_int ; сегмент про◆
                                   ◆едуры
00B9  8E D8                      MOV DS, AX; помещаем в DS
00BB  B4 25                      MOV AH, 25H; функция установ◆
                                   ◆и вектора
00BD  B0 23                      MOV AL, 23H; номер вектора
00BF  CD 21                      INT 21H; меняем прерывание
00C1  1F                      POP DS

00C2                                ctrl_c:
00C2  B4 00                      mov ah, 0
00C4  CD 16                      int 16h
00C6  3C 03                      cmp al, 3 ;код символа посл◆
                                   ◆ нажатия
00C8  75 F8                      jne ctrl_c

00CA  CD 23                      INT 23H

00CC  FA                      CLI
00CD  1E                      PUSH DS
00CE  8B 16 0002 R              MOV DX, KEEP_IP

```

```

00D2  A1 0000 R          MOV AX, KEEP_CS
00D5  8E D8             MOV DS, AX
00D7  B4 25             MOV AH, 25H
00D9  B0 23             MOV AL, 23H
00DB  CD 21             INT 21H

; восстанавливаем стары
й вектор прерывания
00DD  1F               POP DS
00DE  FB               STI

00DF  B4 4C             MOV AH, 4CH
00E1  CD 21             INT 21H

00E3                MAIN ENDP
00E3                CODE ENDS
                END MAIN

```

Microsoft (R) Macro Assembler Version 5.10

12/26/21 17:01:2

Symbols-1

Segments and Groups:

	N a m e	Length	Align	Combine Class
ASTACK	0400	PARA	STACK
CODE	00E3	PARA	NONE
DATA	0004	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
CTRL_C	L NEAR	00C2	CODE

ISTACK	L WORD	0009	CODE	Length = 001E
KEEP_AX	L WORD	0007	CODE	
KEEP_CS	L WORD	0000	DATA	
KEEP_IP	L WORD	0002	DATA	
KEEP_SP	L WORD	0005	CODE	
KEEP_SS	L WORD	0003	CODE	
MAIN	F PROC	009F	CODE	Length = 0044
MY_INT	F PROC	0000	CODE	Length = 009F
START	L NEAR	0066	CODE	
TIME	N PROC	0045	CODE	Length = 0021
WHILE_TIME	L NEAR	005E	CODE	
@CPU	TEXT	0101h		
@FILENAME	TEXT	A		
@VERSION	TEXT	510		

116 Source Lines

116 Total Lines

20 Symbols

48044 + 459216 Bytes symbol space free

0 Warning Errors

0 Severe Errors