

**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. 1383

Сапожников А.Э.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Разработка собственного прерывания.

### **Задание.**

Разработка собственного прерывания, генерируемое при нажатии клавиш Control+C; выдача звукового сигнала с заданной длительностью звучания.

### **Выполнение работы.**

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. Далее программой считывается символ с клавиатуры (int 16h) и когда он равен '3' вызывается прерывание.

В коде прерывания включается воспроизведение звука (out 42h), а также используется таймер который отсчитывает время которое будет воспроизводиться звук (int 15h/ah=86h по значениям регистров cx,dx). Все изменяемые регистры хранятся в стеке, и возвращают своё значение перед возвратом из прерывания.

В конце программы восстанавливается старый вектор прерывания

Таблица 1- Тестирование программы

№	Входная строка	Выходная строка	Комментарий
1	<ctrl+c>	*beep*<2s>	Всё правильно
2	123QWE	-	Вывода нет, программа не упала
3	123qwe<ctrl+c>	*beep*<2s>	Звучит 2 секунды после ctrl+c

Программный код см. в приложении А.

### **Выводы.**

Разработана программу с собственным прерыванием воспроизводящая звук при нажатии <ctrl+c>.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
; Стек программы
AStack SEGMENT STACK
    DW 1024 DUP(?) ; Отводится 1024 слов памяти
AStack ENDS

; Данные программы
DATA SEGMENT
    KEEP_CS DW 0 ; для хранения сегмента
    KEEP_IP DW 0 ; и смещения прерывания
DATA ENDS

; Код программы
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack

BEEP PROC FAR
    PUSH AX ; сохранение изменяемых регистров
    PUSH CX
    PUSH DX
; действия по обработке прерывания

    MOV AL,10110110B ; Установка
    OUT 43H,AL ; режима таймера

    IN AL,61H ; Состояние динамика
    OR AL,00000011b ; Включение
    OUT 61H, AL ; динамика

    MOV AX, 2712 ;установка частоты звука
    OUT 42H,AL
    MOV AL,AH
    OUT 42H,AL
    MOV CX,001eh
    mov dx,8480h
;задержка
    mov ah,86h
    int 15h ;15h - 86H прерывание ожидания
;CX,DX = сколько микросекунд ждать
    IN AL,61H ; Выключение динамика
    AND AL,11111100B
    OUT 61H,AL
```

; восстановление регистров

POP DX

POP CX

POP AX

MOV AL, 20H

OUT 20H, AL

IRET

BEEP ENDP

; Головная процедура

Main PROC FAR

PUSH DS

sub AX,AX

PUSH AX

mov AX,DATA

mov DS,AX

MOV AH, 35H

MOV AL, 23H

INT 21H

MOV KEEP\_IP, BX

MOV KEEP\_CS, ES

PUSH DS

MOV DX, OFFSET BEEP

MOV AX, SEG BEEP

MOV DS, AX

MOV AH, 25H

MOV AL, 23H

INT 21H

POP DS

skipnotcntrlc:

mov ah, 0

int 16h

cmp al, 3

jne skipnotcntrlc

int 23h

CLI

PUSH DS

MOV DX, KEEP\_IP

MOV AX, KEEP\_CS

MOV DS, AX

MOV AH, 25H

MOV AL, 23H

INT 21H

```
Main    ENDP
CODE    ENDS
        END Main
```

Page 1-1

```

; Ժլ՜՜,ԺմԺ° Ժլ՜՜ Ժ՜Ժ³՜՜ Ժ°ԺԸԺԸ՜՜
0000          AStack      SEGMENT STACK
0000 0400[      DW 1024 DUP(?) ; Ժ ՜՜,Ժ²Ժ՜Ժ՜Ժ՜՜՜,՜ ՜՜
          1024 ՜՜ Ժ»Ժ՜Ժ² ԺլԺ°ԺԸ՜՜ ՜՜,Ժ՜
          ???
          ]

0800          AStack      ENDS

; Ժ՜՜Ժ°ԺԸԺԸ՜՜՜Ժմ Ժլ՜՜ Ժ՜Ժ³՜՜ Ժ°ԺԸԺԸ՜՜՜
0000          DATA      SEGMENT
0000 0000      KEEP_CS DW 0 ; Ժ՜Ժ»՜՜ ՜՜...՜՜ Ժ°ԺԸԺմԺԸԺ՜՜
՜՜ ՜՜ ԺմԺ³ԺԸԺմԺԸ՜՜,Ժ°
0002 0000      KEEP_IP DW 0 ; Ժ՜ ՜՜ ԺԸԺմ՜՜՜ԺմԺԸԺ՜՜՜ Ժ
լ՜՜ Ժմ՜ ՜՜՜Ժ²Ժ°ԺԸԺ՜՜՜
0004          DATA      ENDS

; Ժ՜Ժ՜Ժ՜Ժ Ժլ՜՜ Ժ՜Ժ³՜՜ Ժ°ԺԸԺԸ՜՜՜
0000          CODE      SEGMENT
          ASSUME CS:CODE, DS:DATA, SS:AStack

0000          BEEP PROC FAR
0000 50          PUSH AX ; ՜՜ Ժ՜՜՜...՜՜ Ժ°ԺԸԺմԺԸԺ՜՜Ժմ Ժ՜Ժ·ԺԸԺ
մԺԸ՜՜ ԺմԺԸ՜՜՜՜... ՜՜ ԺմԺ³Ժ՜՜ ՜՜,՜՜ Ժ՜Ժ²
0001 51          PUSH CX
0002 52          PUSH DX
; Ժ՜ԺմԺ¹՜՜ ՜՜,Ժ²Ժ՜՜ ԺլԺ՜ Ժ՜Ժ±՜՜ Ժ°Ժ±Ժ՜՜,Ժ°Ժմ Ժլ՜՜
Ժմ՜ ՜՜՜Ժ²Ժ°ԺԸԺ՜՜՜

0003 B0 B6          MOV AL,10110110B ; Ժլ՜ ՜՜,Ժ°ԺԸԺ՜Ժ²Ժ°Ժ°
0005 E6 43          OUT 43H,AL ; ՜՜ ԺմԺ՜՜Ժ՜ԺԸԺ° ՜՜,Ժ°Ժ¹Ժ
ԸԺմ՜ ՜՜°

0007 E4 61          IN AL,61H ; ԺլԺ՜՜ ՜՜,Ժ՜՜ ԺԸԺ՜Ժմ Ժ

```

```

                                ŽŹĐœĐ°ĐœĐŹĐ°Đ°
0009 0C 03                     OR AL,00000011b ; Đ' Đ°Đ»Ń Ń‡ĐμĐœĐŹĐμ
000B E6 61                     OUT 61H, AL ; ĐŽĐŹĐœĐ°ĐœĐŹĐ°Đ°

000D B8 0A98                   MOV AX, 2712 ;ŃfŃ Ń,Đ°ĐœĐŸĐ²Đ°Đ° Ń‡Đ°Ń
                                Ń,ĐŸŃ,Ń< Đ·Đ²ŃfĐ°Đ°
0010 E6 42                     OUT 42H,AL
0012 8A C4                     MOV AL,AH
0014 E6 42                     OUT 42H,AL
0016 B9 001E                   MOV CX,001eh
0019 BA 8480                   mov dx,8480h
                                ;Đ·Đ°ĐŹĐμŃ ĐŸĐ°Đ°
001C B4 86                     mov ah,86h
001E CD 15                     int 15h ;15h - 86H ĐĹŃ ĐμŃ Ń<Đ²Đ°ĐœĐ
                                ŹĐμ ĐŸĐŸĐŸĐŹĐŹĐ°ĐœĐŹŃ
                                ;CX,DX = Ń Đ°ĐŸĐ»ŃĐ°ĐŸ ĐœĐŹĐ°Ń ĐŸŃ ĐμĐ
                                °ŃfĐœĐŹ ĐŸĐŹĐ°Ń,ŃĚ
0020 E4 61                     IN AL,61H ; Đ' Ń<Đ°Đ»Ń Ń‡ĐμĐœĐŹĐμ ĐŹĐŹĐœ
                                Đ°ĐœĐŹĐ°Đ°
0022 24 FC                     AND AL,11111100B
Microsoft (R) Macro Assembler Version 5.10 12/18/22 19:51:1
                                Page 1-2

```

```

0024 E6 61                     OUT 61H,AL

                                ;Đ²ĐŸŃ Ń Ń,Đ°ĐœĐŸĐ²Đ»ĐμĐœĐŹĐμ Ń ĐμĐ²ĐŹŃ Ń,Ń ĐŸ
                                Đ²
0026 5A                       POP DX
0027 59                       POP CX
0028 58                       POP AX
0029 B0 20                     MOV AL, 20H
002B E6 20                     OUT 20H, AL
002D CF                       IRET
002E                           BEEP ENDP

                                ; Đ“ĐŸĐ»ĐŸĐ²ĐœĐ°Ń ĐĹŃ ĐŸŃ‡ĐμĐŹŃfŃ Đ°
002E                           Main PROC FAR
002E 1E                       PUSH DS
002F 2B C0                     sub AX,AX
0031 50                       PUSH AX

0032 B8 ---- R                mov AX,DATA
0035 8E D8                     mov DS,AX

0037 B4 35                     MOV AH, 35H
0039 B0 23                     MOV AL, 23H
003B CD 21                     INT 21H

```

```

003D 89 1E 0002 R      MOV KEEP_IP, BX
0041 8C 06 0000 R      MOV KEEP_CS, ES
0045 1E                PUSH DS
0046 BA 0000 R        MOV DX, OFFSET BEEP
0049 B8 ---- R        MOV AX, SEG BEEP
004C 8E D8            MOV DS, AX
004E B4 25            MOV AH, 25H
0050 B0 23            MOV AL, 23H
0052 CD 21            INT 21H
0054 1F                POP DS

0055                  skipnotcntrlc:
0055 B4 00            mov ah, 0
0057 CD 16            int 16h
0059 3C 03            cmp al, 3
005B 75 F8            jne skipnotcntrlc
005D CD 23            int 23h

005F FA                CLI
0060 1E                PUSH DS
0061 8B 16 0002 R      MOV DX, KEEP_IP
0065 A1 0000 R        MOV AX, KEEP_CS
0068 8E D8            MOV DS, AX
006A B4 25            MOV AH, 25H
006C B0 23            MOV AL, 23H
006E CD 21            INT 21H
0070 1F                POP DS
0071 FB                STI
0072 CA FFFE          RET far

```

```

Microsoft (R) Macro Assembler Version 5.10      12/18/22 19:51:1
Page 1-3

```

```

0075      Main  ENDP
0075      CODE  ENDS
          END Main

```

```

Microsoft (R) Macro Assembler Version 5.10      12/18/22 19:51:1
Symbols-1

```

# Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK .....	0800	PARA	STACK
CODE .....	0075	PARA	NONE
DATA .....	0004	PARA	NONE



Symbols:

N a m e	Type	Value	Attr
BEEP .....	F PROC	0000	CODE Length = 002E
KEEP_CS .....	L WORD	0000	DATA
KEEP_IP .....	L WORD	0002	DATA
MAIN .....	F PROC	002E	CODE Length = 0047
SKIPNOTCNTRLC .....	L NEAR	0055	CODE
@CPU .....	TEXT	0101h	
@FILENAME .....	TEXT	lab5	
@VERSION .....	TEXT	510	

96 Source Lines

96 Total Lines

13 Symbols

48004 + 459256 Bytes symbol space free

0 Warning Errors

0 Severe Errors