

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: СОЗДАНИЕ СОБСТВЕННЫХ ПРЕРЫВАНИЙ.

Студент гр. 0383

Преподаватель

Подопригора И.П.

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Вариант 9:

Создать собственное прерывание с номером 60h, вызываемое в программе, которое будет выдавать звуковой сигнал заданной частоты.

Выполнение работы.

Прерывание реализовано в процедуре SUBR_INT. Перед тем, как прерывание будет вызвано, в регистр AX необходимо занести положительное число – высоту звука, чем больше это число, тем ниже звук. Для генерации тональности использован таймер-счетчик 8253. Канал 2 микросхемы 8253 дает выход на динамик. Для установки канала таймера программа посылает код 0B6H в порт 43H, управляющий порт микросхемы 8253. Тем самым канал 2 таймера-счетчика настраивается на работу в качестве делителя частоты. Таймер делит исходную частоту — в данном случае 1.19МГц — на 16-битовое число, которое программа загружает в регистр канала 2. Регистр канала 2 расположен по адресу порта 42H, в который поочередно загружаются 2 байта из AX. Далее биты 0 и 1 управляющего порта 61h устанавливаются в единицу, чтобы разрешить порту 2 таймера управлять динамиком и включить динамик соответственно, и в цикле, проходящем не менее $64 * 1024$ раз выводится звук, после динамик выключается и восстанавливается значения использованных регистров AX и CX, занесенные в стек в начале процедуры.

В процедуре Main с помощью функции 35h/int 21h запоминается текущий вектор прерывания под номером 60h в переменные KEEP_CS, KEEP_IP. С помощью функции 25h/int 21h устанавливается новый вектор прерывания (реализованная процедура прерывания). Далее это прерывание вызывается в программе, предварительно в AX командой MOV заносится некоторое положительное число, соответствующее высоте звука. В конце программы вектор прерывания под номером 60h восстанавливается с помощью переменных KEEP_CS и KEEP_IP.

Таблица 1. Проверка работы программы.

Команды, вызываемые в программе для проверки работы прерывания.	Результат работы программы
MOV AX, 3000 INT 60H	Выводится звук
MOV AX, 1000 INT 60H	Выводится более высокий звук, чем в первом случае
MOV AX, 8000 INT 60H	Выводится более низкий звук, чем в первом случае

Тексты исходных файлов программ см. в приложении А.

Тексты файлов диагностических сообщений см. в приложении Б.

Выводы.

В ходе выполнения данной лабораторной работы была изучена разработка собственных прерываний на языке Ассемблер.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: **lr5.asm**

; Стек программы

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

STACK SEGMENT STACK

 DW 1024 DUP (?)

STACK ENDS

DATA SEGMENT

 KEEP_CS DW 0 ; для хранения сегмента

 KEEP_IP DW 0 ; и смещения прерывания

DATA ENDS

CODE SEGMENT

SUBR_INT PROC FAR

 PUSH AX ; сохранение изменяемых регистров

 PUSH CX

 ;<действия по обработке прерывания>

 MOV CX, AX

 MOV AL, 10110110b

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

 MOV AX, CX ; Заносим в AX высоту звука

 OUT 42H, AL

 MOV AL, AH

 OUT 42H, AL ; Заносим поочередно 2 байта в порт 42h

 IN AL, 61H

 MOV AH, AL

 OR AL, 3

 OUT 61H, AL ; Установление битов 0 и 1 в единицу

 SUB CX, CX

 WORK_TIME:

 NOP

 NOP

 NOP

 NOP

 NOP

 LOOP WORK_TIME ; Цикл, пока динамик работает

 MOV AL, AH

 OUT 61H, AL ; Выключение динамика (изначальное значение порта 61h)

 ;<конец действий по обработке прерывания>

 POP CX

 POP AX ; восстановление регистров

 MOV AL, 20H

 OUT 20H, AL

 IRET

SUBR_INT ENDP

Main PROC FAR

; <Запоминание текущего вектора прерывания>

MOV AH, 35H ; функция получения вектора

MOV AL, 60H ; номер вектора

INT 21H

MOV KEEP_IP, BX ; запоминание смещения

MOV KEEP_CS, ES ; и сегмента

; <Установка вектора прерывания>

PUSH DS

MOV DX, OFFSET SUBR_INT ; смещение для процедуры в DX

MOV AX, SEG SUBR_INT ; сегмент процедуры

MOV DS, AX ; помещаем в DS

MOV AH, 25H ; функция установки вектора

MOV AL, 60H ; номер вектора

INT 21H ; меняем прерывание

POP DS

MOV AX, 8000

INT 60H

; <Восстановление изначального вектора прерывания>

CLI

PUSH DS

MOV DX, KEEP_IP

MOV AX, KEEP_CS

MOV DS, AX

MOV AH, 25H

MOV AL, 60H

INT 21H ; восстанавливаем вектор

POP DS

STI

MOV AH, 4Ch

INT 21h

Main ENDP

CODE ENDS

END Main

ПРИЛОЖЕНИЕ Б **ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ**

Название файла: **lr5.lst**

Microsoft (R) Macro Assembler Version 5.10

11/10/21 22:56:3

Page 1-1

```

                                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 ; для хранения
                                сегмента
0002 0000                                KEEP_IP DW 0 ; и смещения прер
                                ывания
0004                                DATA    ENDS

0000                                CODE     SEGMENT

0000                                SUBR_INT PROC FAR
0000 50                                PUSH AX    ; сохранение изИ
                                1/4еняемых регистров
0001 51                                PUSH CX

                                ;<действия по обрабИ
                                3/4тке прерывания>
0002 8B C8                                MOV CX, AX
0004 B0 B6                                MOV AL, 10110110b
0006 E6 43                                OUT 43H, AL ; Код для устанИ
                                3/4вления канала 2 таймера-сэ
                                1/4етчика на работу в качест
                                ве делителя частоты
0008 8B C1                                MOV AX, CX ; Заносим в AX выз
                                1/4оту звука
000A E6 42                                OUT 42H, AL
000C 8A C4                                MOV AL, AH
000E E6 42                                OUT 42H, AL ; Заносим пооче

```

	редно 2 байта в порт 42h
0010 E4 61	IN AL, 61H
0012 8A E0	MOV AH, AL
0014 0C 03	OR AL, 3
0016 E6 61	OUT 61H, AL ; Установление
	битов 0 и 1 в единицу
0018 2B C9	SUB CX, CX
001A	WORK_TIME:
001A 90	NOP
001B 90	NOP
001C 90	NOP
001D 90	NOP
001E 90	NOP
001F E2 F9	LOOP WORK_TIME ; Цикл, пока дИ
	,намик работает
0021 8A C4	MOV AL, AH
0023 E6 61	OUT 61H, AL ; Выключение ди
	намика (изначальное значе

Microsoft (R) Macro Assembler Version 5.10

11/10/21 22:56:3

Page 1-2

	ние порта 61h)
	; <конец действий по
	обработке прерывания>
0025 59	POP CX
0026 58	POP AX ; восстановленияИ
	μ регистров
0027 B0 20	MOV AL, 20H
0029 E6 20	OUT 20H, AL
002B CF	IRET
002C	SUBR_INT ENDP
002C	Main PROC FAR
	; <Запоминание текущеИ
	³о вектора прерывания>
002C B4 35	MOV AH, 35H ; функция полэ
	□ чения вектора
002E B0 60	MOV AL, 60H ; номер вектора
0030 CD 21	INT 21H
0032 89 1E 0002 R	MOV KEEP_IP, BX ; запоминание
	смещения
0036 8C 06 0000 R	MOV KEEP_CS, ES ; и сегмента

```

; <Установка вектора прерывания>
003A 1E          PUSH DS
003B BA 0000 R    MOV DX, OFFSET SUBR_INT ; смещение
                  для процедуры в DX
003E B8 ---- R    MOV AX, SEG SUBR_INT ; сегмент И
                  для процедуры
0041 8E D8        MOV DS, AX ; помещаем в
DS
0043 B4 25        MOV AH, 25H ; функция усе
                  жения вектора
0045 B0 60        MOV AL, 60H ; номер вект
3/4ра
0047 CD 21        INT 21H ; меняем преэ
                  жидование
0049 1F          POP DS

004A B8 1F40      MOV AX, 8000
004D CD 60        INT 60H

```

```

; <Восстановление изначального вектора прерывания>
004F FA          CLI
0050 1E          PUSH DS
0051 8B 16 0002 R MOV DX, KEEP_IP
0055 A1 0000 R    MOV AX, KEEP_CS
0058 8E D8        MOV DS, AX
005A B4 25        MOV AH, 25H
005C B0 60        MOV AL, 60H
005E CD 21        INT 21H ; восстанавли
2аем вектор

```

Microsoft (R) Macro Assembler Version 5.10

11/10/21 22:56:3

Page 1-3

```

0060 1F          POP DS
0061 FB          STI

0062 B4 4C        MOV AH, 4Ch
0064 CD 21        INT 21h
0066             Main ENDP
0066             CODE ENDS
                  END Main

```


Segments and Groups:

N a m e	Length	Align	Combine	Class
CODE	0066	PARA		NONE
DATA	0004	PARA		NONE
STACK	0800	PARA		STACK

Symbols:

N a m e	Type	Value	Attr	
KEEP_CS	L WORD	0000	DATA	
KEEP_IP	L WORD	0002	DATA	
MAIN	F PROC	002C	CODE	Length = 003A
SUBR_INT	F PROC	0000	CODE	Length = 002C
WORK_TIME	L NEAR	001A	CODE	
@CPU	TEXT	0101h		
@FILENAME	TEXT	lr5		
@VERSION	TEXT	510		

85 Source Lines

85 Total Lines

13 Symbols

48034 + 459226 Bytes symbol space free

0 Warning Errors

0 Severe Errors