

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»
Тема: Написание собственного прерывания.

Студент гр. 0383

Позолотин К.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Разработать программу, устанавливающую собственное прерывание.

Задание.

Номер вектора прерывания: 60h - прерывание пользователя - должно генерироваться в программе. Необходима выдача звукового сигнала с заданной длительностью звучания.

Порядок выполнения работы.

Программа начинается с процедуры main. Затем происходит запоминание вектора прерывания по номеру 60h, при помощи прерывания 21h ah = 35h. Затем происходит установка пользовательского прерывания, при помощи процедуры subr_int. Прерывание устанавливается на выполнение процедуры proc_r. Используется регистр ds, поэтому его значение запоминается в стеке. Для вывода звука сначала вызывается процедура включения звукового сигнала. Затем процедура задержки и уже потом процедура прекращения звукового сигнала. В конце программы выполняется восстановление старого прерывания.

Процедура задержки работает за счет проверки на каждой итерации, сменилось ли значение секунд в текущий момент. Если секунда прошла, то значение delay уменьшается на 1.

Процедура включения звука задаёт частоту звука передавая в порт 42h значения bx и включая динамик изменяя значение крайних двух бит.

Вывод.

Были разработана программа для создания собственного вектора прерывания, которая включает динамик на определённое время.

ТЕСТИРОВАНИЕ

При запуске программа издаёт звуковой сигнал по длине равный заданному в программе количеству секунд.

Приложение А

Тексты исходных файлов программы

Название файла: lr5.asm

data segment

seconds db 120

delay db 0

keep_cs dw 0

keep_ip dw 0

data ends

astack segment stack

dw 512 dup(0)

astack ends

code segment

assume cs:code, ss:astack, ds:data

sound_start proc near

push ax

mov al, 10110110b ;цепочка для командного регистра 8253

out 43h, al ;посылаем в регистр

mov ax, bx

out 42h, al ;посылаем младший байт

mov al, ah ;посылаем в регистр

```
out    42h, al    ;посылаем старший байт
```

```
in     al,61h     ;Получить состояние динамика
```

```
or     al,03h     ;Установить бит
```

```
out    61h,al     ;Включить динамик
```

```
pop    ax
```

```
ret
```

```
sound_start endp
```

```
sound_end proc near
```

```
push   ax
```

```
in     al,61h     ;Получить состояние динамика
```

```
and    al,0fch    ;Установить бит
```

```
out    61h,al     ;Выключить динамик
```

```
pop    ax
```

```
ret
```

```
sound_end endp
```

```
proc_delay proc near
```

```
push   ax
```

```
push   dx
```

delaying:

; получаем системное время

mov ah, 2ch

int 21h ;

cmp dh, seconds ; сравниваем текущие секунды с seconds

je delaying

mov seconds, dh ; сохраняем секунды

dec delay ; вычитаем 1 из задержки

jnz delaying ; если задержка не равна 0, повторяем

pop dx

pop ax

ret

proc_delay endp

proc_r proc near

push bx

push cx

mov bx, 20000

call sound_start

mov delay, ah

call proc_delay

call sound_end

pop cx

pop bx

```

        iret;
proc_r endp

subr_int proc near
        ; устанавливаем новый вектор прерывания

        mov al,60h ; номер вектора
        mov ah,25h ; функция установки вектора
        push ds
        mov dx, offset proc_r ; смещение для процедуры в dx
        mov ax, seg proc_r ; сегмент процедуры
        mov ds, ax ; перемещаем в ds
        mov ax, 2560h ; функция установки флага(25) и номер вектора (60)
        int 21h ; меняем прерывание
        pop ds
        ret
subr_int endp

main proc far
        mov ax, ds ; загрузка сегментного
        mov ds, ax ; регистра данных

        mov ah, 35h ; функция получения вектора
        mov al, 60h ; номер вектора
        int 21h ; возвращает текущее значение вектора прерывания
        mov keep_ip, bx ; запоминание смещения
        mov keep_cs, es ; и сегмента вектора прерывания

```

```
call subr_int
```

```
mov ah, 10; устанавливается задержка
```

```
int 60h;
```

```
    ; восстанавливаем старый вектор прерывания
```

```
cli ; отключение аппаратных прерываний
```

```
push ds
```

```
mov dx, keep_ip
```

```
mov ax, keep_cs
```

```
mov ds, ax
```

```
mov ah, 25h ; функция установки вектора
```

```
mov al, 1ch ; номер вектора
```

```
int 21h
```

```
pop ds
```

```
sti ; установка прерываний
```

```
mov ax, 4c00h ; завершение программы
```

```
int 21h
```

```
main endp;
```

```
code ends
```

```
end main
```

Название файла: lab5.lst


```
0000          data segment
0000 78          seconds db 120
0001 00          delay  db 0

0002 0000          keep_cs dw 0
0004 0000          keep_ip dw 0
0006          data ends

0000          astack segment stack
0000 0200[          dw 512 dup(0)
          0000
          ]

0400          astack ends

0000          code segment
          assume cs:code, ss:astack, ds:data
0000          sound_start proc near
0000 50              push  ax

0001 B0 B6              mov  al, 10110110b  ;цепочка для к
                                омандного регистра 8253
0003 E6 43              out  43h, al      ;посылаем в ре
                                гистр
0005 8B C3              mov  ax, bx
```

0007	E6 42	out 42h, al	;посылаем млад
		ший байт	
0009	8A C4	mov al, ah	;посылаем в ре
		гистр	
000B	E6 42	out 42h, al	;посылаем стар
		ший байт	
000D	E4 61	in al,61h	;Получить состо
		яние динамика	
000F	0C 03	or al,03h	;Установить бит
0011	E6 61	out 61h,al	;Включить динам
		ик	
0013	58	pop ax	
0014	C3	ret	
0015		sound_start endp	
0015		sound_end proc near	
0015	50	push ax	
0016	E4 61	in al,61h	;Получить состо
		яние динамика	

```
0018 24 FC          and    al,0fch    ;Установить бит
001A E6 61          out     61h,al    ;Выключить дина
                                мик
```

```
001C 58            pop     ax
001D C3            ret
001E              sound_end endp
```

```
001E              proc_delay proc near
001E 50            push    ax
001F 52            push    dx
0020              delaying:
                                ; получаем системное время
0020 B4 2C          mov     ah, 2ch
0022 CD 21          int     21h ;
0024 3A 36 0000 R   cmp     dh, seconds ; сравниваем текущие секунды
                                с secodns
0028 74 F6          je     delaying
002A 88 36 0000 R   mov     seconds, dh ; сохраняем секунды
002E FE 0E 0001 R   dec     delay ; вычитаем 1 из задержки
0032 75 EC          jnz     delaying ; если задержка не равна 0, по
                                вторяем
```

0034	5A	pop	dx
0035	58	pop	ax
0036	C3	ret	
0037		proc_delay	endp
0037		proc_r	proc near
0037	53	push	bx
0038	51	push	cx
0039	BB 4E20	mov	bx, 20000
003C	E8 0000 R	call	sound_start
003F	88 26 0001 R	mov	delay, ah
0043	E8 001E R	call	proc_delay
0046	E8 0015 R	call	sound_end
0049	59	pop	cx
004A	5B	pop	bx
004B	CF	iret;	
004C		proc_r	endp
004C		subr_int	proc near
			; устанавливаем новый вектор прерывания
004C	B0 60	mov	al, 60h ; номер вектора

```
004E B4 25          mov ah,25h ; функция установки вектора
0050 1E             push ds
0051 BA 0037 R       mov dx, offset proc_r ; смещение для процедур
                     в dx
0054 B8 ---- R       mov ax, seg proc_r ; сегмент процедуры
0057 8E D8           mov ds, ax ; перемещаем в ds
0059 B8 2560         mov ax, 2560h ; функция установки флага(25) и
                     номер вектора (60)
005C CD 21           int 21h ; меняем прерывание
005E 1F             pop ds
005F C3             ret
0060               subr_int endp

0060               main proc far
0060 8C D8           mov ax, ds ; загрузка сегментного
0062 8E D8           mov ds, ax ; регистра данных

0064 B4 35          mov ah, 35h ; функция получения вектора
0066 B0 60          mov al, 60h ; номер вектора
0068 CD 21          int 21h ; возвращает текущее значение вектор
                     а прерывания
006A 89 1E 0004 R    mov keep_ip, bx ; запоминание смещения
006E 8C 06 0002 R    mov keep_cs, es ; и сегмента вектора прерыва
                     ния
```

0072	E8 004C R	call subr_int
0075	B4 0A	mov ah, 10; устанавливается задержка
0077	CD 60	int 60h;
		; восстанавливаем старый вектор прерываний
		ия
0079	FA	cli ; отключение аппаратных прерываний
007A	1E	push ds
007B	8B 16 0004 R	mov dx, keep_ip
007F	A1 0002 R	mov ax, keep_cs
0082	8E D8	mov ds, ax
0084	B4 25	mov ah, 25h ; функция установки вектора
0086	B0 1C	mov al, 1ch ; номер вектора
0088	CD 21	int 21h
008A	1F	pop ds
008B	FB	sti ; установка прерываний
008C	B8 4C00	mov ax, 4c00h ; завершение программы
008F	CD 21	int 21h
0091		main endp;
0091		code ends
		end main

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0400	PARA		STACK
CODE	0091	PARA		NONE
DATA	0006	PARA		NONE

Symbols:

N a m e	Type	Value	Attr	
DELAY	L BYTE	0001	DATA	
DELAYING	L NEAR	0020	CODE	
KEEP_CS	L WORD	0002	DATA	
KEEP_IP	L WORD	0004	DATA	
MAIN	F PROC	0060	CODE	Length = 0031
PROC_DELAY	N PROC	001E	CODE	Length = 0019
PROC_R	N PROC	0037	CODE	Length = 0015
SECONDS	L BYTE	0000	DATA	
SOUND_END	N PROC	0015	CODE	Length = 0009
SOUND_START	N PROC	0000	CODE	Length = 0015

SUBR_INT N PROC 004C CODE Length = 0014

@CPU TEXT 0101h

@FILENAME TEXT 1ab5

@VERSION TEXT 510

141 Source Lines

141 Total Lines

19 Symbols

48014 + 459246 Bytes symbol space free

0 Warning Errors

0 Severe Errors

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММ

Файл LAB5.asm

