

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.

Студентка гр. 1381

Новак П.И.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить работу с прерываниями на языке Ассемблер и создать собственное.

Задание.

Вариант 17: 3с

23h -прерывание, генерируемое при нажатии клавиш Control+C.

C - выдача звукового сигнала с заданной длительностью звучания.

Выполнение работы.

Программа начинается с процедуры main. Сначала в стеке сохраняется значение сегмента даты и регистра ax, затем происходит проверка на заданное в программе время, если оно ненулевое, происходит считывание ctrl+c. В случае ввода символа не с кодовым значением 3, программа снова запрашивает ввод при помощи прерывания 16h. Затем происходит запоминание вектора прерывания 23h с помощью функции 35 прерывания 21h. В keep_cs и keep_ip записываются сегмент и смещение. После этого с помощью функции 25h прерывания 21h происходит установка пользовательского прерывания и его вызов.

SOUND - процедура пользовательского прерывания, в которой включается звуковой динамик. В переменной time хранится время, которое требуется для проигрывания звукового сигнала. Изначально в метке sound_start происходит установка константной частоты, включение динамика за счёт установки битов в 0 и 1. Время работы динамика контролируется в метке timer: на каждом шаге проверяется, изменилось ли значение секунд, если да, то значение переменной time уменьшается на 1. По истечению времени программа переходит к метке sound_end, где биты обнуляются, проигрывание останавливается.

В конце программы восстанавливается оригинальный вектор прерывания, из стека возвращается значение ds. Исходный код программы смотреть в приложении А.

Файл листинг смотреть в приложении Б.

Выводы.

В ходе работы были изучены прерывания языка ассемблер. Создано собственное прерывание, которое по нажатой клавише выдает звуковой сигнал с заданной длительностью звучания.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab5.asm

```
DATA SEGMENT
```

```
sec db 120
```

```
time db 2
```

```
keep_cs dw 0 ;для хранения сегмента
```

```
keep_ip dw 0 ;и смещения прерывания
```

```
DATA ENDS
```

```
AStack SEGMENT STACK
```

```
DW 512 DUP(?)
```

```
AStack ENDS
```

```
CODE SEGMENT
```

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

```
SOUND PROC FAR ;обработка прерывания
```

```
push ax
```

```
sound_start:
```

```
    mov al, 10110110b
```

```
    out 43h, al
```

```
    mov ax, 4400
```

```
    ;устанавливаем частоту
```

```
    out 42h, al
```

```
    mov al, ah
```

```
    out 42h, al
```

```
    ;включаем динамик
```

```
    in al, 61h ;текущее состояние порта 61h в AL
```

```
    or al, 00000011b ;устанавливаем биты 0 и 1 в 1 (разрешаем работу динамика и включить его)
```

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

```

inc time
timer:
    mov ah, 2ch
    int 21h
    cmp dh, sec
    je timer

mov sec, dh
dec time
jnz timer

sound_end;;выключаем динамик
    in al, 61h
    and al, 11111100b ;обнуляем младшие два бита
    out 61h, al
    pop ax
    mov al, 20h
    out 20h, al
    iret
SOUND ENDP

Main PROC FAR
push ds
sub ax, ax
push ax
mov ax, DATA
mov ds, ax

cmp time,0
jne read_symbol

read_symbol:
mov ah, 0
int 16h
cmp al, 3 ;

```

```

jne read_symbol

mov ax, 3523h
int 21h
mov keep_ip, bx ;запоминание смещения
mov keep_cs, es ;и сегмента
push ds
mov dx, offset SOUND ;смещение для процедура в DX
mov ax, seg SOUND ;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
int 23h ;вызов прерывания

CLI;восстанавливаем старый вектор прерывания
push ds
mov dx, keep_ip
mov ax, keep_cs
mov ds, ax
mov ah, 25h
mov al, 23h
int 21h ;восстанавливаем вектор
pop ds
STI
RET

Main ENDP
CODE ENDS
END Main

```


ПРИЛОЖЕНИЕ Б

ЛИСТИНГ

Название файла: lab5.lst

Microsoft (R) Macro Assembler Version 5.10
22:41:3

11/14/22

Page

1-1

0000

DATA

SEGMENT

0000 78

sec db

120

0001 02

time db

2

0002 0000

keep_cs

dw 0 ;для хранения сегм
ента

0004 0000

keep_ip

dw 0 ;и смещения прерыв
ания

0006

DATA

ENDS

0000

AStack

SEGMENT STACK

0000 0200[

DW 512

DUP (?)

????

]

0400		
		AStack
ENDS		
0000		
		CODE
SEGMENT		
ASSUME CS:CODE, DS:DATA, SS:AStack		
0000		
		SOUND
PROC FAR ;обработка прерыв		
ания		
0000 50		push ax
0001		
		sound_st
art:		
0001 B0 B6		mov
al, 10110110b		
0003 E6 43		out
43h, al		
0005 B8 1130		mov
ax, 4400		
;устанавливаем частоту		
0008 E6 42		out
42h, al		
000A 8A C4		mov
al, ah		
000C E6 42		out
42h, al		
;включаем динамик		

000E	E4 61	
	61h ;текущее состояние	in al,
	порта 61h в AL	
0010	0C 03	
	00000011b ;устанавливаем	or al,
	биты 0 и 1 в 1 (разрешаем раб	
	оту динамика и включить ей	
	³o)	
0012	E6 61	
	61h, al ;включаем динамик	out
0014	FE 06 0001 R	inc
	time	
0018		
		timer:
0018	B4 2C	
	ah, 2ch	mov
001A	CD 21	
	21h	int
001C	3A 36 0000 R	
	cmp dh, sec	
0020	74 F6	
	timer	je
0022	88 36 0000 R	
	mov sec, dh	
0026	FE 0E 0001 R	
	dec time	
002A	75 EC	
	timer	jnz
002C		
		sound_en
	d:;выключаем динамик	
002C	E4 61	
		in al,

61h

Microsoft (R) Macro Assembler Version 5.10
22:41:3

11/14/22

Page

1-2

002E 24 FC

and

al, 11111100b ;обнуляем младэ

ие два бита

0030 E6 61

out

61h, al

0032 58

pop ax

0033 B0 20

mov

al, 20h

0035 E6 20

out

20h, al

0037 CF

iret

0038

SOUND

ENDP

0038

Main

PROC FAR

0038 1E

push ds

0039 2B C0

sub ax,

ax

003B 50

push ax

003C B8 ---- R

mov ax, DATA

003F 8E D8

mov ds,

ax

```
0041  80 3E 0001 R 00                                cmp
time,0

0046  75 00                                              jne
read_symbol

0048

read_sym

bol:
0048  B4 00                                              mov ah,
0
004A  CD 16                                              int 16h
004C  3C 03                                              cmp al,
3 ;
004E  75 F8                                              jne
read_symbol

0050  B8 3523                                              mov ax,
3523h
0053  CD 21                                              int 21h

0055  89 1E 0004 R
mov keep_ip, bx ;запоминание сме
щения
0059  8C 06 0002 R
mov keep_cs, es ;и сегмента
005D  1E
push ds

005E  BA 0000 R
mov dx, offset SOUND ;смещение для
процедура в DX
0061  B8 ---- R
mov ax, seg SOUND ;сегмент процед
уры
0064  8E D8                                              mov ds,
ax ;помещаем в DS
```

```

0066  B4 25                                mov ah,
25h ;функция установки
вектора
0068  B0 23                                mov al,
23h ;номер вектора
006A  CD 21                                int 21h
;меняем прерывание
006C  1F                                    pop ds
006D  CD 23                                int 23h
;вызов прерывания

006F  FA
CLI;восстанавливаем старый
^ вектор прерывания
0070  1E                                    push ds

0071  8B 16 0004 R
mov dx, keep_ip
0075  A1 0002 R
mov ax, keep_cs
0078  8E D8                                mov ds,
ax
007A  B4 25                                mov ah,
25h
007C  B0 23                                mov al,
23h
007E  CD 21                                int 21h
;восстанавливаем век
тор
Microsoft (R) Macro Assembler Version 5.10    11/14/22
22:41:3
Page
1-3

```

```

0080  1F                                     pop ds
0081  FB                                     STI
0082  CB                                     RET

0083

Main
ENDP
0083

CODE
ENDS
END Main
Microsoft (R) Macro Assembler Version 5.10    11/14/22
22:41:3

Symbols-1

```

Segments and Groups:

Name	Length	Align	Combine
Class			
ASTACK	0400		PARA STACK
CODE	0083		PARA NONE
DATA	0006		PARA NONE

Symbols:

Name	Type	Attr
Value		

KEEP_CS	L WORD 0002 DATA
KEEP_IP	L WORD 0004 DATA
MAIN	F PROC 0038 CODE Length =
004B	
READ_SYMBOL	L NEAR 0048 CODE
SEC	L BYTE 0000 DATA
SOUND	F PROC 0000 CODE Length =
0038	
SOUND_END	L NEAR 002C CODE
SOUND_START	L NEAR 0001 CODE
TIME	L BYTE 0001 DATA
TIMER	L NEAR 0018 CODE
@CPU 0101h	TEXT
@FILENAME lab5	TEXT
@VERSION 510	TEXT

98 Source Lines

98 Total Lines

18 Symbols

47942 + 432678 Bytes symbol space free

0 Warning Errors

0 Severe Errors

