

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.

Студентка гр. 0382

Михайлова О.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить прерывания на языке Ассемблер и создать собственное.

Задание.

Вариант 24: 4с

4 - 16h - прерывание от клавиатуры (по заданному скан-коду клавиши делать действия A-G, указанные ниже).

C - выдача звукового сигнала с заданной длительностью звучания.

Выполнение работы.

Программа начинается с процедуры main. Сначала считывается скан-код клавиши, введенной пользователем, при помощи прерывания 16h. Считывание происходит до тех пор, пока пользователь не нажмет клавишу "C". Затем происходит запоминание вектора прерывания 16h с помощью функции 35 прерывания 21h. В keep_cs и keep_ip записываются сегмент и смещение соответственно. После этого с помощью функции 25h прерывания 21h происходит установка пользовательского прерывания и его вызов.

SUBR_INT - процедура пользовательского прерывания, в которой включается звуковой динамик. Он работает определенное количество времени, записанное в переменной time в сегменте данных, и затем выключается. Время работы динамика контролируется в метке timer: на каждом шаге проверяется, изменилось ли значение секунд, если да, то значение переменной time уменьшается на 1.

В конце программы восстанавливается оригинальный вектор прерывания.

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 5
    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

SUBR_INT 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

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

    ;выключаем динамик
    sound_end:
        in al, 61h
        and al, 11111100b ;обнуляем младшие два бита
        out 61h, al

    pop ax
    mov al, 20h
    out 20h, al
    iret
SUBR_INT ENDP
```

```

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

read_symbol:
    mov ah, 0
    int 16h
    cmp ah, 46 ;скан-код клавиши "C"
    jne read_symbol

    mov ah, 35h ;функция получения вектора
    mov al, 16h ;номер вектора
    int 21h
    mov keep_ip, bx ;запоминание смещения
    mov keep_cs, es ;и сегмента

    push ds
    mov dx, offset SUBR_INT ;смещение для процедура в DX
    mov ax, seg SUBR_INT ;сегмент процедуры
    mov ds, ax ;помещаем в DS
    mov ah, 25h ;функция установки вектора
    mov al, 16h ;номер вектора
    int 21h ;меняем прерывание
    pop ds

    int 16h

    ;восстанавливаем старый вектор прерывания
    CLI
    push ds
    mov dx, keep_ip
    mov ax, keep_cs
    mov ds, ax
    mov ah, 25h
    mov al, 16h
    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
22:36:5

12/12/21

Page

1-1

```
0000          DATA SEGMENT
0000  78          sec db 120
0001  05          time db 5
0002  0000          keep_cs dw 0 ;PrP»CЦ
C...CЪP°PSPµPSPëCЦ CГPµP
0004  0000          iPjPµPSC,P°
PiCЪPµCЪC          keep_ip dw 0 ;Pë CГPjPµC%PµPSPëCЦ
0006          <PIP°PSPëCЦ
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          SUBR_INT PROC FAR ;PsP±CЪP°P±PsC,PeP°
PiCЪPµCЪC          <PIP°PSPëCЦ
0000  50          push ax

0001          sound_start:
0001  B0 B6          mov al, 10110110b
0003  E6 43          out 43h, al
0005  B8 1130          mov ax, 4400
          ;CГCГC,P°PSP°P»PëPIP°PµPj CГP°CГC,PsC,C
          í
0008  E6 42          out 42h, al
000A  8A C4          mov al, ah
000C  E6 42          out 42h, al
          ;PIPeP»CЪC±P°PµPj PrPëPSP°PjPëPe
000E  E4 61          in al, 61h ;C,PµPeCГC%PµPµ
CГPScГC          ,PsCЦPSPëPµ PiPsCЪC,P° 61h PI AL
0010  0C 03          or al,
00000011b ;CГCГC,P°PSP°PIP»Pë
          PIP°PµPj P±PëC,C< 0 Pë 1 PI 1
(CЪP°P·CЪPµCЕР°Pµ
```

```

Pj CbP°P±PsC,Cr PrPëPSP°PjPëPeP° Pë
PIPeP»CtC+P
0012 E6 61 out 61h, al ;PIPeP»CtC+P°PµPj
PrPëPSP°P
jPëPe

0014 timer:
0014 B4 2C mov ah, 2ch
0016 CD 21 int 21h
0018 3A 36 0000 R cmp dh, sec
001C 74 F6 je timer
001E 88 36 0000 R mov sec, dh
0022 FE 0E 0001 R dec time
0026 75 EC jnz timer

;PIC<PeP»CtC+P°PµPj PrPëPSP°PjPëPe
0028 sound_end:
0028 E4 61 in al, 61h

```

Microsoft (R) Macro Assembler Version 5.10
22:36:5

12/12/21

Page

1-2

```

002A 24 FC and al,
11111100b ;PsP±PSCrP»CµPµPj P
jP»P°PrCëPëPµ PrPIP° P±PëC,P°
002C E6 61 out 61h, al

002E 58 pop ax
002F B0 20 mov al, 20h
0031 E6 20 out 20h, al
0033 CF iret
0034 SUBR_INT ENDP

0034 Main PROC FAR
0034 1E push ds
0035 2B C0 sub ax, ax
0037 50 push ax
0038 B8 ---- R mov ax, DATA
003B 8E D8 mov ds, ax

003D read_symbol:
003D B4 00 mov ah, 0
003F CD 16 int 16h
0041 80 FC 2E cmp ah, 46 ;CfPeP°PS-PePsPr
PeP»P°PIPëC
ëPë "C"
0044 75 F7 jne read_symbol

0046 B4 35 mov ah, 35h ;C,,CfPSPeC†PëCµ
PiPsP»CfC+PµPSP
ëCµ PIPµPeC,PsCbP°
0048 B0 16 mov al, 16h ;PSPsPjPµCb
PIPµPeC,PsCbP°
004A CD 21 int 21h

```

```

004C 89 1E 0004 R      mov keep_ip,
bx ;P·P°PìPsPjPëPSP°PSPëPµ CÍP
jPµC%PµPSPëCµ
0050 8C 06 0002 R      mov keep_cs, es ;Pë CÍPµPìPjPµPSC,P°

0054 1E                push ds
0055 BA 0000 R          mov dx, offset SUBR_INT ;CÍPjPµC%PµPSPëPµ
P
rP»Cµ PìCßPsC†PµPrCíCßP° Pí DX
0058 B8 ---- R          mov ax, seg SUBR_INT ;CÍPµPìPjPµPSC,
PìCßPs
C†PµPrCíCßC<
005B 8E D8              mov ds, ax ;PìPsPjPµC%P°PµPj Pí DS
005D B4 25              mov ah, 25h ;C„CíPSPeC†PëCµ
CíCÍC,P°PSPsPìP
ePë PìPµPeC,PsCßP°
005F B0 16              mov al, 16h ;PSPsPjPµCß
PìPµPeC,PsCßP°
0061 CD 21              int 21h ;PjPµPSCµPµPj
PìCßPµCßC<PìP°PSPëPµ
0063 1F                pop ds
0064 CD 16              int 16h

;PìPsCÍCÍC,P°PSP°PìP»PëPìP°PµPj
CÍC,P°CßC<P
N° PìPµPeC,PsCß PìCßPµCßC<PìP°PSPëCµ
0066 FA                CLI
0067 1E                push ds
0068 8B 16 0004 R        mov dx, keep_ip
006C A1 0002 R          mov ax, keep_cs
006F 8E D8              mov ds, ax
0071 B4 25              mov ah, 25h

```

Microsoft (R) Macro Assembler Version 5.10
12/12/21 22:36:5

Page

1-3

```

0073 B0 16              mov al, 16h
0075 CD 21              int
21h ;PìPsCÍCÍC,P°PSP°PìP»PëPìP°PµPj PìP
µPeC,PsCß
0077 1F                pop ds
0078 FB                STI
0079 CB                ret

007A                    Main ENDP
007A                    CODE ENDS
END Main

```

bols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0400	PARA	STACK
CODE	007A	PARA	NONE
DATA	0006	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
	KEEP_CS	L WORD	0002	DATA
	KEEP_IP	L WORD	0004	DATA
0046	MAIN	F PROC	0034	CODE Length =
	READ_SYMBOL	L NEAR	003D	CODE
	SEC	L BYTE	0000	DATA
	SOUND_END	L NEAR	0028	CODE
	SOUND_START	L NEAR	0001	CODE
0034	SUBR_INT	F PROC	0000	CODE Length =
	TIME	L BYTE	0001	DATA
	TIMER	L NEAR	0014	CODE
	@CPU	TEXT	0101h	
	@FILENAME	TEXT	lab5	
	@VERSION	TEXT	510	

98 Source Lines
98 Total Lines
18 Symbols

48014 + 459246 Bytes symbol space free

0 Warning Errors
0 Severe Errors