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. 0382 _____ Дудко М.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

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 set_int. Прерывание устанавливается на выполнение процедуры R. Используется регистр ds, поэтому его значение запоминается в стэке. 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
ИСХОДНЫЙ КОД ПРОГРАММ

Файл LAB5.asm

```
data segment
    seconds  db 120
    delay    db 5  ;how many seconds to wait.

    keep_cs dw 0 ; store seg
    keep_ip dw 0 ; offset
data ends

astack segment stack
    dw 512 dup(0)
astack ends

code segment
    assume cs:code, ss:astack, ds:data
beepstart proc near
     push  ax

     mov     al, 10110110b

     out      43h, al           ; send it to the initializing port
43h timer 2.

     mov     ax, bx

  ;SOUND
     out     42h, al          ; send lsb to port 42h.
     mov     al, ah           ; move msb into al
     out     42h, al          ; send msb to port 42h.

  ;ON
  in      al,61h          ;see value of port 61h
  or      al,03h          ;set bits 0 and 1
  out     61h,al          ;update port 61h

     pop   ax
  ret
beepstart endp

beepend proc near
     push  ax

  ;OFF
  in      al,61h          ;see value of port 61h
  and     al,0fch         ;reset bits 0 and 1
  out     61h,al          ;update port 61h

     pop   ax
  ret
beepend endp

my_delay proc near
push  ax
push  dx
delaying:
;get system time.
```

```
  mov  ah, 2ch
  int  21h ;return seconds in dh.
;check if one second has passed.
  cmp  dh, seconds
  je   delaying
;if no jump, one second has passed. very important : preserve
;seconds to use them to compare with next seconds. this is how
;we know one second has passed.
  mov  seconds, dh
  dec  delay
  jnz  delaying  ;if delay is not zero, repeat.

pop  dx
pop  ax

  ret
my_delay endp

r proc near ; interupt
      push bx
      push cx

      mov bx, 20000        ; rate
      call beepstart

      mov delay, ah; time
      call my_delay
      call beepend

      pop cx
      pop bx

      iret;
r endp

set_int proc near
  mov al,60h;
  mov ah,25h;
  push ds
  mov dx, offset r
  mov ax, seg r
  mov ds, ax
  mov ax, 2560h
  int 21h
  pop ds
  ret
set_int endp

main proc far
  mov  ax, ds
  mov  ds, ax     ;initialize data segment.

;----------get vec---------
  mov  ah, 35h   ; get vec
  mov  al, 60h   ; num vec
  int  21h
  mov  keep_ip, bx  ; offset
  mov  keep_cs, es  ; seg
```

```
;---------my vec---------
  call set_int
  mov ah, 5; delay in secodns +- 1 second :c
  int 60h;


;-----Re vec-------
  cli
  push ds
  mov  dx, keep_ip
  mov  ax, keep_cs
  mov  ds, ax
  mov  ah, 25h
  mov  al, 1ch
  int  21h
  pop  ds
  sti

  mov ax, 4c00h ; выход в dos
  int 21h

main endp;
code ends
end main
```

Файл LAB5.lst

```
data segment
 0000  78               seconds  db 120
 0001  05               delay    db 5  ;how many seconds to wait.


 0002  0000                   keep_cs dw 0 ; store seg
 0004  0000                   keep_ip dw 0 ; offset
 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               beepstart proc near
 0000  50                push  ax


 0001  B0 B6                  mov     al, 10110110b
```

```
      0003  E6 43                   out     43h, al          ; send it
to t
                        he initializing port 43h timer 2.

      0005  8B C3                   mov     ax, bx

                          ;SOUND
      0007  E6 42                   out     42h, al          ; send lsb
to
                        port 42h.
      0009  8A C4                   mov     al, ah           ; move msb
int
                        o al
      000B  E6 42                   out     42h, al          ; send msb
to
                        port 42h.

                          ;ON
      000D  E4 61                     in      al,61h          ;see value of
port 61
                        h
      000F  0C 03                     or       al,03h          ;set bits 0
and 1
      0011  E6 61                     out     61h,al          ;update port
61h

      0013  58                      pop   ax
      0014  C3                    ret
      0015                    beepstart endp

      0015                    beepend proc near
      0015  50                      push  ax

                          ;OFF
      0016  E4 61                     in      al,61h          ;see value of
port 61
                        h
      0018  24 FC                     and     al,0fch         ;reset bits 0
and 1
```

```
      001A  E6 61                out      61h,al           ;update port
61h

     Microsoft      (R)      Macro      Assembler      Version      5.10
11/14/22 12:00:3
                                                                    Page
1-2


      001C  58                     pop   ax
      001D  C3                   ret
      001E                    beepend endp

      001E                    my_delay proc near
      001E  50                push ax
      001F  52                push dx
      0020                    delaying:
                              ;get system time.
      0020  B4 2C                    mov  ah, 2ch
      0022  CD 21                    int  21h ;return seconds in dh.
                              ;check if one second has passed.
      0024  3A 36 0000 R             cmp  dh, seconds
      0028  74 F6                    je   delaying
                              ;if  no  jump,  one  second  has  passed.  very
import
                              ant : preserve
                              ;seconds  to  use  them  to  compare  with  next
secon
                              ds. this is how
                              ;we know one second has passed.
      002A  88 36 0000 R             mov  seconds, dh
      002E  FE 0E 0001 R             dec  delay
      0032  75 EC                     jnz   delaying  ;if delay is not zero,
repeat.

      0034  5A                 pop   dx
      0035  58                 pop   ax

      0036  C3                   ret
      0037                    my_delay endp
```

```
0037                r proc near ; interupt
0037  53                  push bx
0038  51                  push cx

0039  BB 4E20                mov bx, 20000        ; rate
003C  E8 0000 R           call beepstart

003F  88 26 0001 R              mov delay, ah; time
0043  E8 001E R           call my_delay
0046  E8 0015 R           call beepend

0049  59                  pop cx
004A  5B                  pop bx

004B  CF                  iret;
004C                r endp

004C                set_int proc near
004C  B0 60                 mov al,60h;
004E  B4 25                 mov ah,25h;
0050  1E              push ds
0051  BA 0037 R       mov dx, offset r
0054  B8 ---- R       mov ax, seg r
0057  8E D8                 mov ds, ax
0059  B8 2560               mov ax, 2560h
```

```
005C  CD 21                int 21h
005E  1F              pop ds
005F  C3              ret
0060                set_int endp

0060                main proc far
0060  8C D8                mov  ax, ds
0062  8E D8                 mov   ds,  ax        ;initialize  data
segment.

                    ;----------get vec---------
0064  B4 35                mov  ah, 35h   ; get vec
0066  B0 60                mov  al, 60h   ; num vec
0068  CD 21                int  21h
006A  89 1E 0004 R         mov  keep_ip, bx  ; offset
006E  8C 06 0002 R         mov  keep_cs, es  ; seg


                    ;---------my vec---------
0072  E8 004C R       call set_int
0075  B4 05                 mov  ah,  5;  delay  in  secodns  +-  1
second :c
0077  CD 60                int 60h;


                    ;-----Re vec-------
0079  FA              cli
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
0086  B0 1C                mov  al, 1ch
0088  CD 21                int  21h
```

```
008A  1F              pop  ds
008B  FB              sti

008C  B8 4C00               mov ax, 4c00h ; выход в dos
008F  CD 21                 int 21h

0091                main endp;
0091                code ends
                    end main
```

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
BEEPEND	N PROC	0015	CODE Length = 0009
BEEPSTART	N PROC	0000	CODE Length = 0015
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
MY_DELAY	N PROC	001E	CODE Length = 0019
R	N PROC	0037	CODE Length = 0015
SECONDS	L BYTE	0000	DATA
SET_INT	N PROC	004C	CODE Length = 0014

```
@CPU . . . . . .     TEXT  0101h
@FILENAME  . . . . . .     TEXT  LAB5
@VERSION . . . . . .     TEXT  510


    137 Source  Lines
    137 Total   Lines
     19 Symbols

  48014 + 459246 Bytes symbol space free

      0 Warning Errors
 0 Severe  Errors
```