МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №5
по дисциплине «Организация систем и ЭВМ»
Тема «Написание собственного прерывания.»

Студент гр. 1303 _____ Кропотов Н.Д.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Изучить особенности прерываний на языке Ассемблера, написать собственное прерывание.

Задание.

Вариант 1С.

Написать прерывание 08h – прерывание от системного таймера генерируется автоматически операционной системой 18 раз в сек. Реализовать выдачу звукового сигнала с заданной длительностью звучания.

Выполнение работы.

Объявляются два двухбайтовых сегмента памяти SAVE_IP и SAVE_CS. Они используются для сохранения смещения до оригинального прерывания и позволяют в дальнейшем восстановить исходные вектора прерывания.

Описывается процедура SUBR_INT, которая является написанным пользовательским прерыванием. В данной процедуре в начале все регистры, которые будут изменены, для сохранения кладутся в стек, затем осуществляется взаимодействие с динамиком компьютера – выставляется частота звука, время звучания, сохраняется состояние порта, биты, отвечающие за доступ к динамику и его включение выставляются в 1.

После проигрывания звука порт возвращается в исходное состояние. Все сохраненные регистры вынимаются из стека, а также обеспечивается разрешение прерываний более низкого уровня во время действия данного.

В главной процедуре смещение и сегмент прерывания, которое требуется заменить сохраняются в объявленные сегменты памяти. Функция 35H прерывания 21H дает вектор прерывания, записанного в нижний байт регистра AX. Смещение и сегмент данного регистра записываются в регистры BX и ES, соответственно, они сохраняются в SAVE_IP и SAVE_CS. Далее записывается новое прерывание.

Функция 25H считывает смещение до него из DX и сегмент из DS и устанавливает его в вектор прерывания. Так как прерывание 08H вызывается 18 раз в секунду, то для того, чтобы наблюдать результат выполнения пользовательского прерывания, используется зацикливание, которое можно прервать нажатием клавиши Esc.

После выхода из цикла, исходный вектор прерывания восстанавливается, и программа завершается. Исходный код программы см. в приложении А.

Вывод.

В ходе выполнения лабораторной работы были изучены виды прерываний и работа с ними. В соответствии с задание было разработано собственное прерывание.

ПРИЛОЖЕНИЕ А

Тексты исходных файлов программ lab5.asm.

```
DATA SEGMENT
    SAVE_IP DW 0
    SAVE_CS DW 0
DATA ENDS

AStack SEGMENT STACK
    DW 1024 DUP(?)
AStack ENDS

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

SUBR_INT PROC FAR
    push AX
    push BX
    push DX
    push CX

    mov bx, 5   ; sound frequency
    mov al, 0b6h
    out 43h, al
    mov dx, 0014h
    mov ax, 4f38h
    div di

    ; set freq
    out 42h, al
    mov al, ah
    out 42h, al

    ; sound on
    in al, 61H  ; cur port state to al
    mov ah, al  ; save state in ah
    or al, 3    ; set 0 and 1 bit at 1
```

```
    out 61H, al ; speaker on

    l1: mov cx, 2801h
    l2: loop l2
    dec bx
    jnz l1
    mov al, ah
    and al, 11111100b
    out 61H, al


    pop AX
    pop BX
    pop DX
    pop CX
    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

    mov di, 10000

    MOV AH, 35H
    MOV AL, 08H
    INT 21H
    MOV SAVE_IP, BX
    MOV SAVE_CS, ES

    push DS
```

```
    mov DX, offset SUBR_INT
    mov ax, seg SUBR_INT
    mov ds, ax
    mov ah, 25h
    mov al, 08H
    int 21h
    pop ds


loop_int:
    push ax
    mov ah, 1h
    int 21H

    cmp al, "-"
    je minus

    cmp al, "+"
    je plus

    cmp al, 1bh

    pop ax
    jnz loop_int
    jmp exit

plus:
    add di, 1000
    pop ax
    jmp loop_int

minus:
    sub di, 1000
    pop ax
    jmp loop_int
```

```
exit:
    CLI
    PUSH DS
    MOV  DX, SAVE_IP
    MOV  AX, SAVE_CS
    MOV  DS, AX
    MOV  AH, 25H
    MOV  AL, 08H
    INT  21H
    POP  DS
    STI

    ret
Main ENDP

CODE ENDS
     END Main
```