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

ОТЧЁТ

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

Студент гр. 1381 _____ Хомутинников Н.А.

Преподаватель _____ Ефремов М. А.

Санкт-Петербург
2022

Цель работы

Разработать собственное прерывание на языке ассемблера.

Задание

Цифра в шифре задает номер и назначение заменяемого вектора прерывания:

1 - 08h - прерывание от системного таймера – генерируется автоматически операционной системой 18 раз в сек;

Буква определяет действия, реализуемые программой обработки прерываний:

B - Выдача звукового сигнала с заданной высотой звука.

Выполнение работы

1) Задаются следующие константы: *interrupt EQU 08h* – прерывание от системного таймера, *tone EQU 10* – частота звука, *duration EQU 200* – длительность звучания звука. В сегменте данных создаются 2-байтовые переменные для хранения сегмента и смещения прерывания - *keep_cs DW 0* и *keep_ip DW 0*
2) Процедура SOUND_ON PROC FAR включает таймер, устанавливает состояние динамика, частота и длительность звука. Так же имеется вложенный цикл для задержки воспроизведения звука через динамик
3) Процедура RESTORE PROC NEAR восстанавливает изначальные векторы прерываний
4) REPLACE PROC NEAR - функция 35 прерывания 21H возвращает текущее значение вектора прерывания и помещает в es значение сегмента и смещение сегмента в bx. Для задания адреса собственного прерывания с заданным номером в таблицу векторов прерываний используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес нового обработчика

Вывод

Реализовано собственное прерывание на языке ассемблера.

Приложение А

Исходный код программы

Название файла: lab5.asm

```
interrupt EQU 08h
tone EQU 10
duration EQU 200

ASTACK SEGMENT STACK
  DB 2048 DUP(?)
ASTACK ENDS

DATA SEGMENT

keep_cs DW 0
keep_ip DW 0
check DW 300

DATA ENDS

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

MAIN PROC FAR
    push ds
    xor ax, ax
    push ax
    mov ax, DATA
    mov ds, ax
    call REPLACE

repeat:
    cmp BYTE PTR [check], 0
    jne repeat
    call RESTORE
    ret
MAIN ENDP

SOUND_ON PROC FAR
    push ax
    push cx
    mov al, 20h
    out 20h, al
    dec WORD PTR [check]
    mov al, tone
    out 42h, al
    in al, 61h
    or al, 00000011b
    out 61h, al
    mov cx, duration

sound:
    loop sound
    and al, 11111100b
    out 61h, al
    pop cx
    pop ax
```

```
    iret
SOUND_ON ENDP

RESTORE PROC NEAR
    push dx
    push ax

    cli
    push ds
    mov  dx, keep_ip
    mov  ax, keep_cs
    mov  ds, ax
    mov  ah, 25h
    mov  al, interrupt
    int  21h
    pop  ds
    sti

    pop ax
    pop dx
    ret
RESTORE ENDP

REPLACE PROC NEAR
    push ax
    push dx

    mov  ah, 35h
    mov  al, 1ch
    int  21h
    mov  keep_ip, bx
    mov  keep_cs, es

    push ds
    mov  dx, OFFSET sound_on
    mov  ax, SEG sound_on
    mov  ds, ax
    mov  ah, 25h
    mov  al, interrupt
    int  21h
    pop  ds

    pop dx
    pop ax
    ret
REPLACE ENDP
CODE ENDS
END MAIN
```