

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»

Студентка гр. 1303

Сырцева Д.Д

Преподаватель

Ефремов М.А

Санкт-Петербург

2022

Цель работы.

Изучить особенности прерываний на языке Ассемблера. Разработать собственное прерывание.

Задание.

Вариант 23.

Написать прерывание 16h – прерывание от клавиатуры. По заданному скан-коду клавиши произвести выдачу звукового сигнала с заданной высотой звука. Отвести программе под стек не менее 1К байт

Выполнение работы.

Сегмент данных – `my_data`:

- 1) `delay dw 2000` – переменная, хранящая нужную задержку.
- 2) `cached_cs dw 0` – переменная для хранения сегмента того прерывания, которое мы заменяем.
- 3) `cached_ip dw 0` – переменная для хранения смещения замененного нами прерывания.

Под стек выделяется 1Кбайт.

В начале программы происходит сохранение адреса старого обработчика прерывания с помощью функции 35h прерывания 21h. Далее устанавливается адрес нового обработчика прерывания с помощью функции 25h прерывания 21h. Далее определена метка `input_loop` – ждать нажатие клавиши, здесь происходит считывание информации из порта клавиатуры (порт 60h). Если в регистре `al` теперь хранится значение 30h, что соответствует скан-коду клавиши В, то вызывается написанное нами прерывание, если нет, то с помощью команды `jne` снова происходит переход на метку `input_loop`. Также в данной метке в регистр `al` переносится значение будущей высоты звука (задается высота 100).

В начале сегмента кода `my_code` определяется процедура пользовательского прерывания – `my_interruption`. В данной процедуре сначала

сохраняются в стеке значения регистров до входа в прерывание. Далее происходит сохранение состояния динамика, включается динамик и таймер для подачи сигналов на динамик, также устанавливается длительность/высота звука. Далее идет цикл `sound_duration` с вложенным циклом `sound_duration_2`. Это реализовано для того, чтобы выдача звука происходила не слишком быстро (также для этого используется инструкция `por`, которая ничего не делает, но занимает определенное время). После выполнения циклов динамик выключается и восстанавливается его прежнее состояние, также восстанавливаются регистры. После чего происходит возврат из прерывания(`iret`).

После выполнения прерывания происходит сброс флага прерывания (`cli`), восстанавливается старый вектор прерывания и происходит установка флага прерывания(`sti`).

Вывод.

В результате лабораторной работы было написано собственное прерывание, которое выдает звуковой сигнал с заданной высотой звука. .

ПРИЛОЖЕНИЕ А

Текст исходного файла программы lab5.

```
assume cs:my_code, ds:my_data, ss:my_stack

my_stack segment stack
    db 1024 dup(0)
my_stack ends

my_data segment
    delay dw 2000
    cached_cs dw 0
    cached_ip dw 0
my_data ends

my_code segment

my_interruption proc far
    push ax
    push dx
    mov dx, ax

    in al, 61h ; Информация о динамике
    push ax
    or al, 00000011b
    out 61h, al ; Включить динамик
    mov al, dl
    out 42h, al ; Включить таймер для подачи
сигналов на динамик
    mov cx, delay

; Задержка
sound_duration:
    push cx
    mov cx, delay
    sound_duration_2:
        nop
        loop sound_duration_2
    pop cx
    loop sound_duration

    pop ax
    and al, 11111100b
    out 61h, al ; Выключить динамик

    pop dx
    pop ax

    mov al, 20h
```

```

        out 20h, al

        iret
my_interruption endp

main proc far
    push ds
    xor ax, ax
    push ax

    mov ax, my_data
    mov ds, ax

    mov ah, 35h                ; сохраняем адрес старого
прерывания
    mov al, 16h
    int 21h
    mov cached_cs, es
    mov cached_ip, bx

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

    input_loop:                   ; ждать нажатие клавиши
        in al, 60h                ; считать информацию из порта
ввода клавиатуры
        cmp al, 30h                ; 30h -- b
        jne input_loop
        mov al, 100
        int 16h

        cli                        ; Сброс флага прерывания

        push ds
        mov dx, cached_ip
        mov ax, cached_cs
        mov ds, ax
        mov ah, 25h
        mov al, 16h
        int 21h                    ; восстанавливаем старый
вектор прерывания
        pop ds

        sti                        ; Установка флага прерывания

        ret

```

```
main endp
```

```
my_code ends  
end main
```