

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

**ОТЧЕТ**  
**по лабораторной работе № 4**  
**по дисциплине «Организация ЭВМ и систем»**  
**ТЕМА: ИСПОЛЬЗОВАНИЕ АППАРАТНЫХ ПРЕРЫВАНИЙ**

Студент гр. 3311

Баймухамедов Р. Р.

Преподаватель

Гречухин М. Н.

Санкт-Петербург

2024

## Цель работы

Цель работы – знакомство с различного вида аппаратными прерываниями и создание собственных подпрограмм обработки прерываний.

## Задание на лабораторную работу

Спросить у пользователя минимальное и максимальное значение временного интервала (от 0 до 10 секунд). Выбрать случайное значение из заданного интервала. С помощью аппаратного прерывания запустить обратный отсчёт, после которого программа должна закрыться.

*прим. можно использовать глобальные переменные и define*

## Текст программы

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>

// global variables
volatile int remaining_time = 0; // left time in seconds
volatile int tick_counter = 0;   // counter of timer ticks
volatile int run_program = 1;
int min_interval_border = 1, max_interval_border = 10;

// Pointer to old interrupt handler
void interrupt (*oldTimerISR)();

// New interrupt handler
void interrupt newTimerISR() {
    tick_counter++;
    if (tick_counter >= 18) { // One second = 18.2 ticks
        tick_counter = 0;
        if (remaining_time > 0) {
            remaining_time--;
            clrscr();
            gotoxy(10, 10);
            printf("%d", remaining_time);
            gotoxy(wherex()-1, wherey());
        }
        if (remaining_time == 0) {
            setvect(0x08, oldTimerISR); // Return old timer ISR to return system
in original state
            run_program = 0;
        }
    }
}

oldTimerISR(); // Call old timer
```

```

}

int main() {
    int min_interval, max_interval;
    int x1=1, y1=5, x2=65, y2=25; // coordinates of window
    clrscr();
    textbackground(0);
    window(x1, y1, x2, y2); // set the window with coordinates (x1, y1) to (x2,
y2)
    // Enter user prompt
    printf("Enter min interval (%d-%d sec): ", min_interval_border,
max_interval_border);
    scanf("%d", &min_interval);
    printf("Enter max interval (%d-%d sec): ", min_interval_border,
max_interval_border);
    scanf("%d", &max_interval);

    // Is user prompt correct
    if (min_interval < min_interval_border || max_interval > max_interval_border
|| min_interval > max_interval) {
        printf("Incorrect entered values. Please try again!\n");
        return 1;
    }

    // Pick random value in interval
    srand(time(0));
    remaining_time = min_interval + (rand() % (max_interval - min_interval + 1));

    printf("Timer set to %d seconds. Enter any key to start timer",
remaining_time);
    getch();

    clrscr();
    gotoxy(10, 10);
    printf("%d", remaining_time);
    gotoxy(wherex()-1, wherey());

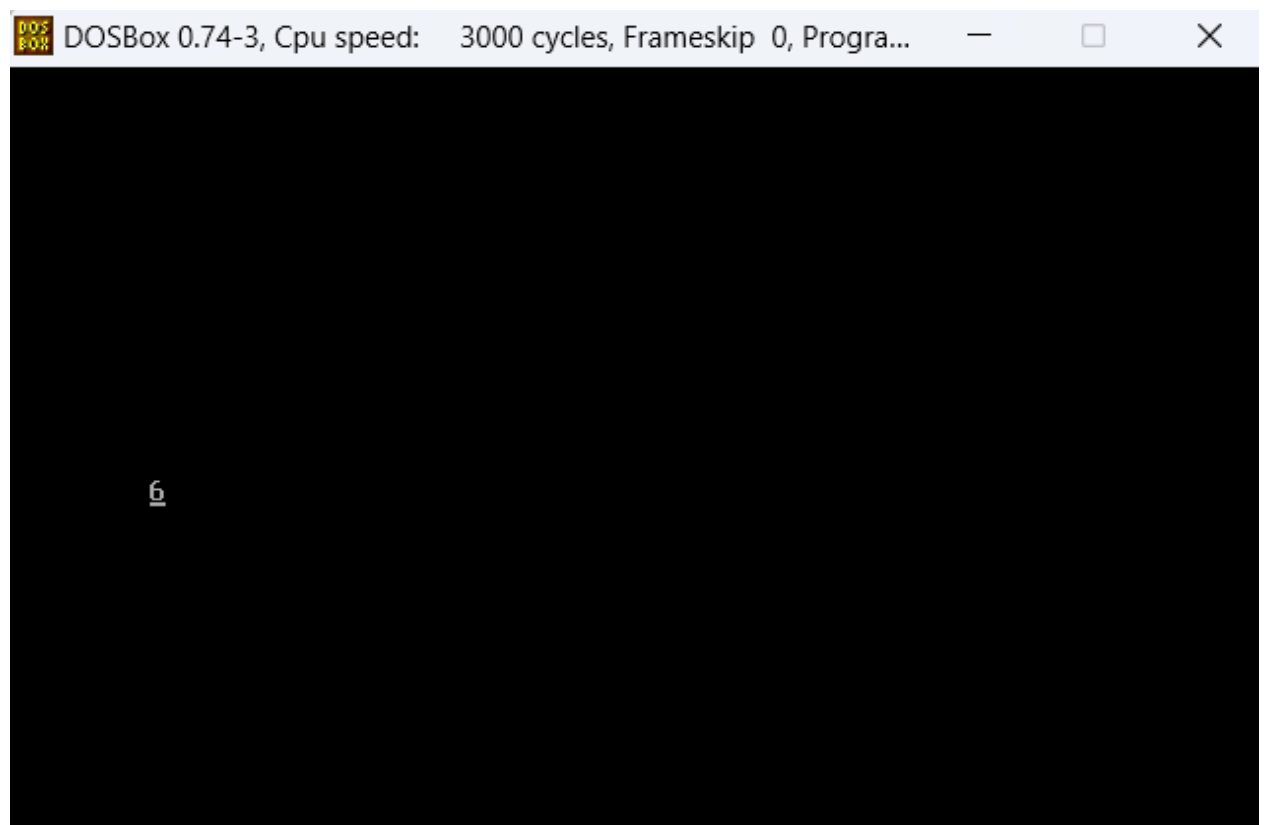
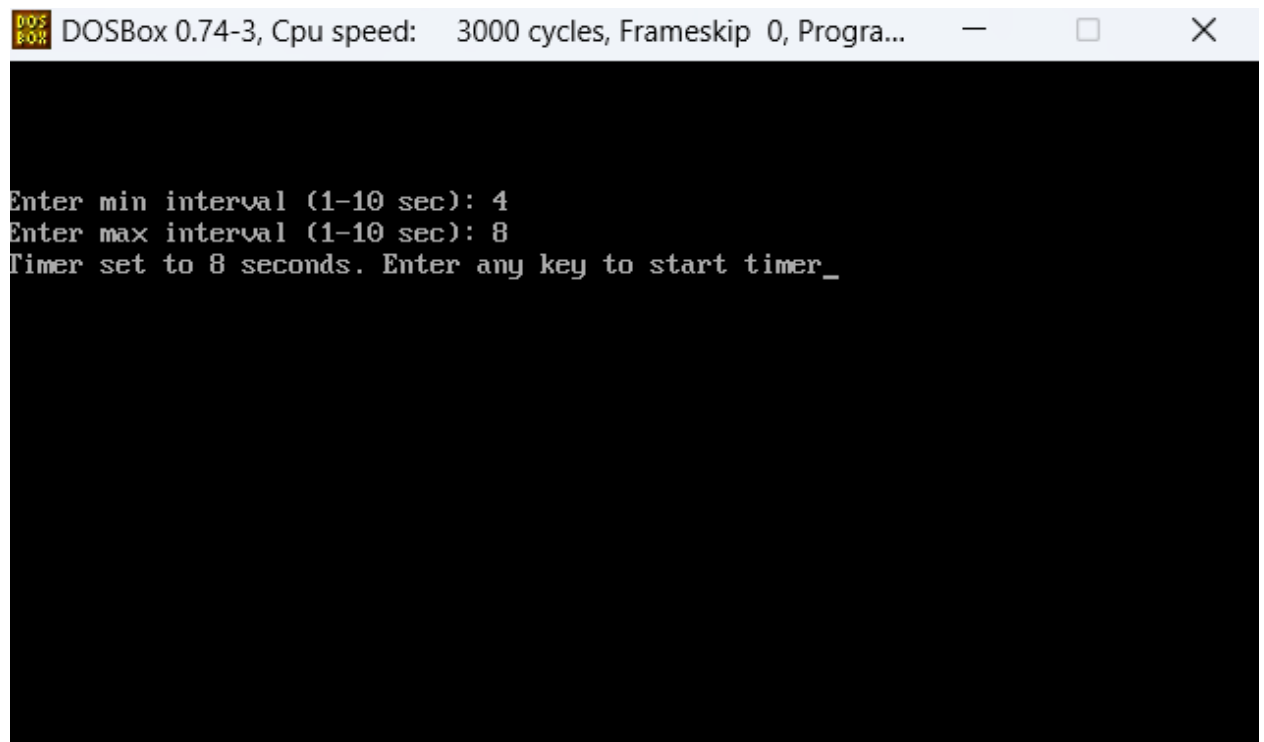
    // Save old timer ISR and set new one
    oldTimerISR = getvect(0x08);
    setvect(0x08, newTimerISR);

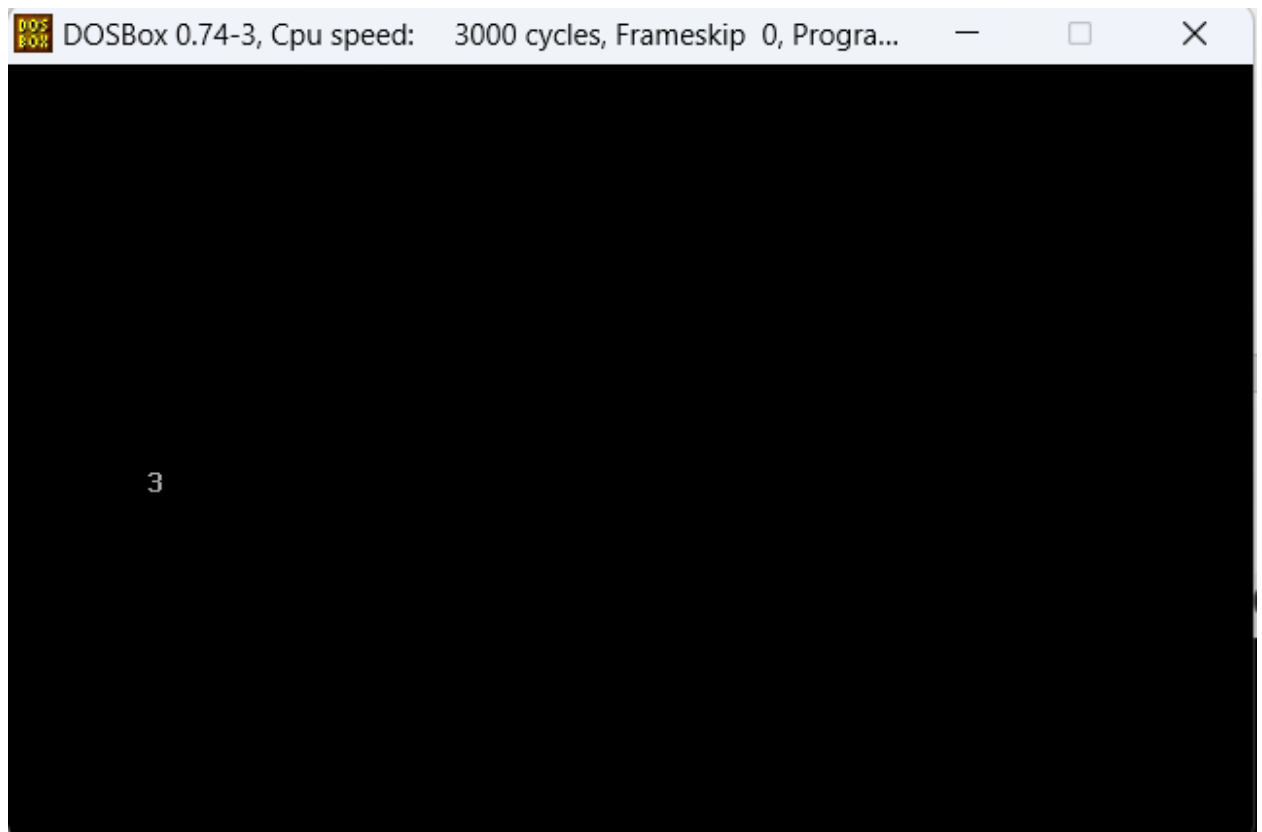
    // Exit from interrupt handler
    while (run_program);

    return 0;
}

```

## Примеры запуска программы





## **Вывод**

В ходе выполнения лабораторной работы я ознакомился с различного вида аппаратными прерываниями, а также создал собственную подпрограмму обработки прерываний