**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 secoonds*

*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 interupt 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 promt*

    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 promt 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);

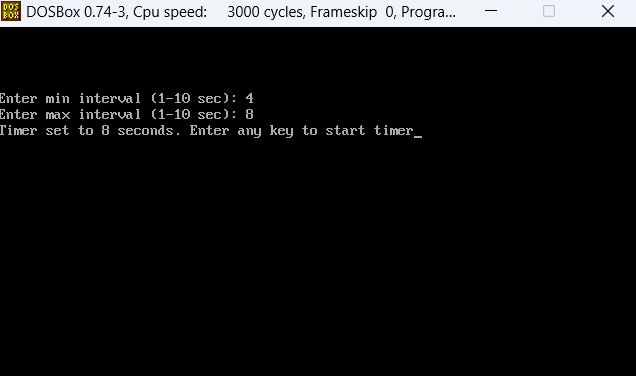
    // Exit from interupt handler

    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