**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Вычислительной техники**

**ОТЧЕТ**

**по лабораторной работе № 5**

**по дисциплине «Организация ЭВМ и систем»**

**Тема: ИСПОЛЬЗОВАНИЕ АППАРАТНЫХ ПРЕРЫВАНИЙ**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3311 |  | Сапронов К. Д. |
| Преподаватель |  | Гречухин М. Н. |

Санкт-Петербург

2024

**Цель работы**

Знакомство с различного вида аппаратными прерываниями и со-здание собственных подпрограмм обработки прерываний.

**Задание**

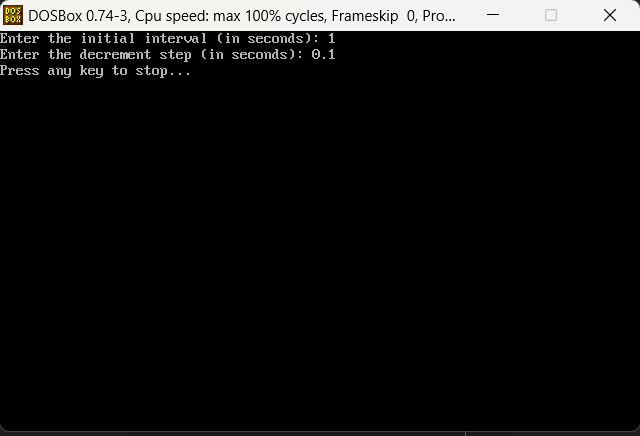
Используя прерывание системного таймера, написать программу, издающую звук с уменьшающимся интервалом. Интервал и шаг уменьшения задается пользователем.

**Описание решения**

Для выполнения задания был написан обрабочик прерывания beep, считывающий значение системного таймера (количество тиков), и если прошло время, равное интервалу или больше интервала, то воспроизводится звук, а значение интервала уменьшается на шаг. Если интервал становится меньше шага, то обработчик перестает издавать звук.

В main программа получает значения интервала и шага от пользователя (они вводятся в секундах, но переводятся в тики для обработчика beep), после чего сохраняется старый обработчик с помощью getvect, и далее он заменяется на beep с помощью setvect. При нажатии любой клавиши обработчик заменяется обратно, и программа завершает свою работу.

**Пример запуска программы**



**Выводы**

Использование аппаратных прерываний в программировании позволяет эффективно перехватывать и изменять поведение устройств в реальном времени, что позволяет выполнять узконаправленные задачи, но требует осторожности, так как ошибки могут привести к нарушению работы других системных компонентов

**Текст программы**

#include <dos.h>

#include <stdio.h>

#include <conio.h>

unsigned long far \*count = (unsigned long far \*)MK\_FP(0x0040, 0x006c);

unsigned int interval, step;

void interrupt beep(...){

    static unsigned long last\_time = 0;

    static unsigned int current\_interval = interval;

    static int active = 1;

    if (\*count - last\_time >= current\_interval && active){

        sound(1000);

        delay(50);

        nosound();

        last\_time = \*count;

        if (current\_interval > step){

        current\_interval -= step;

        } else {

        active = 0;

        }

    }

}

int main(){

    clrscr();

    float interval\_sec, step\_sec;

    printf("Enter the initial interval (in seconds): ");

    scanf("%f", &interval\_sec);

    printf("Enter the decrement step (in seconds): ");

    scanf("%f", &step\_sec);

    interval = (unsigned int)(interval\_sec / 0.055);

    step = (unsigned int)(step\_sec / 0.055);

    if (interval <= step){

        printf("Error: the step must be smaller than the initial interval.\n");

        getch();

        return 1;

    }

    void interrupt (\*oldvect)(...);

    oldvect = getvect(0x1c);

    setvect(0x1c, beep);

    printf("Press any key to stop...\n");

    getch();

    setvect(0x1c, oldvect);

    return 0;

}