**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3353 |  | Карпенко А.Ю. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

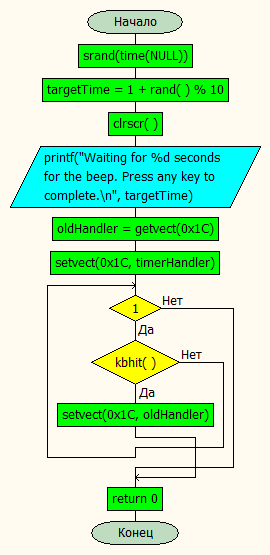
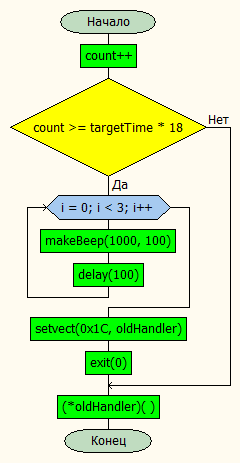
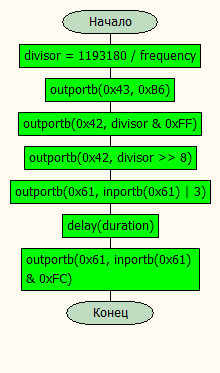
2024

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

**Задание:** По заданию преподавателя необходимо

* Написать программу для работы с прерываниями системного таймера.
* Генерировать случайное количество секунд от 1 до 10.
* По истечении заданного времени произвести 3 звуковых сигнала с использованием системного динамика.
* В случае нажатия клавиши завершить выполнение программы досрочно, восстановив стандартный обработчик прерывания таймера.

**Блок-схема алгоритма**

****

**Текст программы**

#include <dos.h>

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

void interrupt (\*oldHandler)(void); // Указатель на старый обработчик таймера

int count = 0; // Счетчик прерываний таймера

int targetTime; // Случайное количество секунд для отсчета

// Функция для генерации звука с использованием прямого управления динамиком

void makeBeep(int frequency, int duration) {

int divisor = 1193180 / frequency; // Вычисляем делитель для частоты

outportb(0x43, 0xB6); // Настраиваем таймер на генерацию звука

outportb(0x42, divisor & 0xFF); // Устанавливаем младший байт делителя

outportb(0x42, divisor >> 8); // Устанавливаем старший байт делителя

outportb(0x61, inportb(0x61) | 3); // Включаем динамик

delay(duration); // Длительность звука

outportb(0x61, inportb(0x61) & 0xFC); // Выключаем динамик

}

// Обработчик прерывания системного таймера

void interrupt timerHandler(void) {

int i;

count++;

if (count >= targetTime \* 18) { // 18 прерываний примерно соответствуют 1 секунде

targetTime = 1 + rand() % 10; // Сгенерировать новое случайное время

printf("New interval: %d seconds\n", targetTime); // Вывести интервал до сигнала

// Звучание системного динамика 3 раза

for (i = 0; i < 3; i++) {

makeBeep(1000, 100); // Воспроизвести звук с частотой 1000 Гц, длительность 100 мс

delay(100); // Задержка между сигналами 100 миллисекунд

}

count = 0; // Сбросить счетчик

}

(\*oldHandler)(); // Вызвать старый обработчик прерываний

}

int main() {

srand(time(NULL)); // Инициализация генератора случайных чисел

targetTime = 1 + rand() % 10; // Генерация случайного времени от 1 до 10 секунд

clrscr();

// Установка нового обработчика прерываний

oldHandler = getvect(0x1C); // Сохранить старый обработчик прерывания таймера

setvect(0x1C, timerHandler); // Установить новый обработчик

while (!kbhit()) {

// Основной цикл программы, который работает до нажатия любой клавиши

}

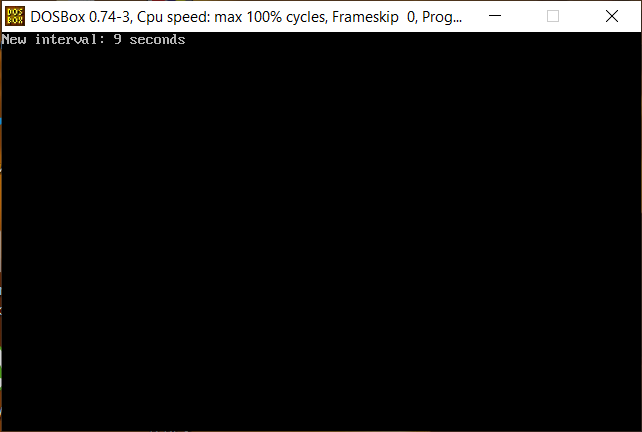
// Восстановление стандартного обработчика перед выходом

setvect(0x1C, oldHandler);

return 0;

}

**Примеры запуска программы**



**Структурная схема аппаратных средств, используемых при выполнении программы с необходимой степенью детализации содержимого блоков.**

1. Монитор (устройство вывода)
2. Клавиатура (устройство ввода)
3. Оперативная память (ОЗУ)
4. Центральный процессор (АЛУ, УУ, Регистры, Кэш-память)

