**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»**

**Вариант 2**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3354 |  | Григорьев В.С. |
| Преподаватель |  | Анисимов А.В. |

Санкт-Петербург

2024

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

**Задание:**

По заданию преподавателя разработать алгоритм и реализовать программу подключения собственной подпрограммы обработки прерывания и использовать её в цепочке со стандартной подпрограммой обработки прерывания от одного из следующих устройств компьютера:

1. системный таймер;

2. клавиатура;

3. контроллер накопителя на гибких магнитных дисках;

4. таймер реального времени;

5.контроллер накопителя на жёстком магнитном диске.

**Блок-схема**

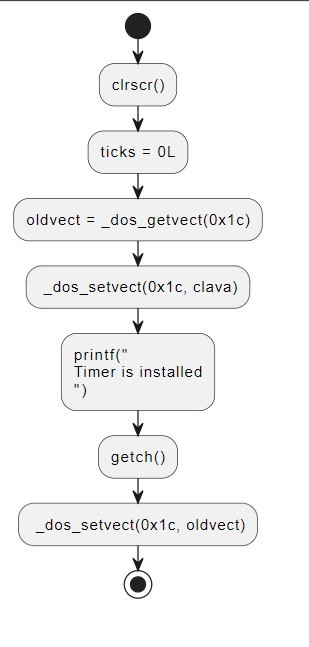


Рисунок 1. Блок-схема

Эта программа написана на языке C и предназначена для работы в среде DOS. Она устанавливает собственный обработчик прерываний для аппаратного прерывания таймера (прерывание 0x1C), которое вызывается примерно 18,2 раза в секунду. Давайте разберём, что делает программа, шаг за шагом:

Объявление переменных и функций:

volatile long ticks; — переменная, которая будет использоваться для подсчета количества срабатываний таймера.

void interrupt far clava(...) — объявление функции, которая будет использоваться в качестве нового обработчика прерывания.

Обработчик прерывания:

В функции clava счетчик ticks увеличивается на 1 каждый раз, когда срабатывает таймер.

Если значение ticks кратно 20 (то есть каждые 20 срабатываний таймера, что соответствует примерно 1 разу в секунду), программа выполняет ассемблерный код, который выводит символ \* на экран.

Затем вызывается старый обработчик прерываний с помощью \_chain\_intr(oldvect), чтобы не нарушить работу других программ, которые также могут использовать таймер.

Основная функция main:

Очищает экран с помощью clrscr().

Сбрасывает счетчик ticks на 0.

Сохраняет адрес старого обработчика прерывания таймера с помощью \_dos\_getvect(0x1c).

Устанавливает новый обработчик прерывания clava для таймера с помощью \_dos\_setvect(0x1c, clava).

Выводит сообщение "Timer is installed".

Ожидает нажатия клавиши с помощью getch().

Восстанавливает старый обработчик прерывания перед завершением программы.

**Текст программы**

#include <dos.h>

#include <stdio.h>

#include <conio.h>

void interrupt(far\* oldvect)(...);

volatile long ticks;

void interrupt far clava(...)

{

ticks++; // Увеличиваем счетчик прерываний таймера

if ((ticks % 20) == 0)

{

asm mov DL, 42

asm mov AH, 2

asm int 21h

} // Если значение счетчика кратно 20, печатаем \*

\_chain\_intr(oldvect); // Вызываем старый обработчик прерывания

}

void main(void)

{

clrscr();

ticks = 0L;// Сбрасываем счетчик

oldvect = \_dos\_getvect(0x1c);// Запоминаем адрес старого обработчика прерывания

\_dos\_setvect(0x1c, clava);// Устанавливаем новый обработчик прерывания

printf("\nTimer is installed\n");

getch();

\_dos\_setvect(0x1c, oldvect); // Восстанавливаем старый обработчик прерывания

}

**Пример работы программы**

****

Рисунок 2. Пример работы программы

**Вывод**: я ознакомился с различного вида аппаратными прерываниями и создал собственную подпрограмму обработки прерываний.