УО БГУИР

Кафедра ЭВМ

Отчет по лабораторной работе

Тема: «Часы реального времени»

Выполнил:

студент группы 650503 Юревич А.С.

Проверил:

к.т.н., доцент Одинец Д.Н.

Минск 2018

1. Постановка задачи

Под MS DOS написать программу, которая:

1) считывает и устанавливает время в часах реального времени;

2) реализует функцию задержки с точностью до миллисекунды;

3) реализует функцию программируемого будильника.

1. Листинг программы

#include <dos.h>

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <io.h>

#include <windows.h>

/\* Время задержки \*/

unsigned int delayTime = 0;

char date[6];

/\* Секунда, минута, час, день, месяц, год \*/

unsigned int registers[] = { 0x00, 0x02, 0x04, 0x07, 0x08, 0x09 };

/\* Старые прерывания \*/

void interrupt(\*oldTimer)(...);

void interrupt(\*oldAlarm) (...);

void getTime();

void setTime();

void delay(unsigned int);

void setAlarm();

void resetAlarm();

void inputTime();

int bcdToDec(int);

int decToBcd(int);

/\* Новые обработчики прерываний \*/

void interrupt newTimer(...)

{

delayTime++;

outp(0x70, 0x0C);

inp(0x71);

/\* Посылаем контроллерам прерываний сигнал end of interruption \*/

outp(0x20, 0x20);

outp(0xA0, 0x20);

}

/\* Новый обработчик для будильника \*/

void interrupt newAlarm(...)

{

puts("Alarm");

oldAlarm();

resetAlarm();

}

int main()

{

int delayMillisecond;

while (1) {

printf("1 - Current time\n");

printf("2 - Set time\n");

printf("3 - Set alarm\n");

printf("4 - Set delay\n");

switch (getch()) {

case '1':

getTime();

break;

case '2':

setTime();

break;

case '3':

setAlarm();

break;

case '4':

fflush(stdin);

printf("Input delay in millisecond: ");

scanf("%d", &delayMillisecond);

delay(delayMillisecond);

break;

default:

system("cls");

break;

}

}

}

/\* Получение текущего времени \*/

void getTime()

{

/\* Названия месяцев \*/

char \*monthToText[] =

{

"January", "Februart", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"

};

system("cls");

int i = 0;

for (i = 0; i < 6; i++)

{

/\* Выбираем нужный регистр \*/

outp(0x70, registers[i]);

/\* Считываем значение из нужного регистра в массив \*/

date[i] = inp(0x71);

}

/\* Переводим считанные значение в десятичную форму \*/

int decDate[6];

for (i = 0; i < 6; i++)

{

decDate[i] = bcdToDec(date[i]);

}

/\* Выводим на экран в нужном порядке \*/

printf(" %2d:%2d:%2d", decDate[2], decDate[1], decDate[0]);

printf(" %s, %2d, 20%2d\n", monthToText[decDate[4] - 1], decDate[3], decDate[5]);

}

/\* Установка времени \*/

void setTime()

{

/\* Вводим новое время \*/

inputTime();

/\* Запрещаем прерывания \*/

disable();

/\* Проверка на доступность значений для чтения и записи \*/

unsigned int res;

do

{

outp(0x70, 0xA);

res = inp(0x71) & 0x80;

} while (res);

/\* Отключаем обновление часов реального времени \*/

outp(0x70, 0xB);

outp(0x71, inp(0x71) | 0x80);

for (int i = 0; i < 4; i++)

{

/\* Выбираем нужный регистр с индексом registers[i]\*/

outp(0x70, registers[i]);

/\* Подаем в него нужное значение \*/

outp(0x71, date[i]);

}

/\* Включаем обновление часов реально времени \*/

outp(0x70, 0xB);

outp(0x71, inp(0x71) & 0x7F);

/\* Разрешаем прерывания \*/

enable();

system("cls");

}

/\* Задержка \*/

void delay(unsigned int ms)

{

/\* Запрещаем прерывания \*/

disable();

/\* Устанавливаем новые обработчики прерываний \*/

oldTimer = getvect(0x70);

setvect(0x70, newTimer);

/\* Разрешаем прерывания \*/

enable();

/\* Размаскирование линии сигнала запроса от ЧРВ \*/

outp(0xA1, inp(0xA1) & 0xFE);

/\* 0xFE = 11111110, бит 0 в 0, чтобы разрешить прерывания от ЧРВ \*/

/\* Выбираем регистр B \*/

outp(0x70, 0xB);

outp(0x71, inp(0x71) | 0x40);

/\* 0x40 = 01000000, 6-й бит регистра B устанавливаем в 1 для периодического прерывания \*/

delayTime = 0;

while (delayTime <= ms);

puts("Delay's end");

setvect(0x70, oldTimer);

return;

}

/\* Установка будильника \*/

void setAlarm()

{

inputTime();

disable();

unsigned int res;

do

{

outp(0x70, 0xA);

res = inp(0x71) & 0x80;

} while (res);

/\* Устанавливаем в регистры будильника нужное время \*/

outp(0x70, 0x05);

outp(0x71, date[2]);

outp(0x70, 0x03);

outp(0x71, date[1]);

outp(0x70, 0x01);

outp(0x71, date[0]);

/\* Выбираем регистр B \*/

outp(0x70, 0xB);

/\* Разрешаем прерывание будильника 5-м битом \*/

outp(0x71, (inp(0x71) | 0x20));

/\* Переопределяем прерывание будильника \*/

oldAlarm = getvect(0x4A);

setvect(0x4A, newAlarm);

outp(0xA1, (inp(0xA0) & 0xFE));

enable();

printf("Alarm enabled\n");

}

/\* Сброс будильника \*/

void resetAlarm()

{

if (oldAlarm == NULL)

{

return;

}

disable();

/\* Возвращаем старое прерывание \*/

setvect(0x4A, oldAlarm);

outp(0xA1, (inp(0xA0) | 0x01));

unsigned int res;

do

{

outp(0x70, 0xA);

res = inp(0x71) & 0x80;

} while (res);

/\* Записываем нулевые значения\*/

outp(0x70, 0x05);

outp(0x71, 0x00);

outp(0x70, 0x03);

outp(0x71, 0x00);

outp(0x70, 0x01);

outp(0x71, 0x00);

outp(0x70, 0xB);

outp(0x71, (inp(0x71) & 0xDF));

enable();

}

void inputTime()

{

int n;

do {

fflush(stdin);

printf("Input hours: ");

scanf("%i", &n);

} while ((n > 23 || n < 0));

date[2] = decToBcd(n);

do {

fflush(stdin);

printf("Input minutes: ");

scanf("%i", &n);

} while (n > 59 || n < 0);

date[1] = decToBcd(n);

do {

fflush(stdin);

printf("Input seconds: ");

scanf("%i", &n);

} while (n > 59 || n < 0);

date[0] = decToBcd(n);

}

int bcdToDec(int bcd)

{

return ((bcd / 16 \* 10) + (bcd % 16));

}

int decToBcd(int dec)

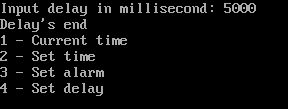
{

return ((dec / 10 \* 16) + (dec % 10));

}}

1. Тестовые пример





1. Заключение

В ходе выполнения работы были получены практические навыки программирования и изучены возможности использования часов реального времени. Был реализован таймер и будильник.