Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчет по лабораторной работе №5

“ Программирование часов реального времени”

Выполнил:

Студент группы 050504 Сидорович Я.А.

Проверил:

к.т.н., доцент Одинец Д.Н.

Минск 2022

1. **Постановка задачи**
2. Написать программу, которая будет считывать и устанавливать время в часах реального времени.
3. Используя аппаратное прерывание часов реального времени и режим генерации периодических прерываний реализовать функцию задержки с точностью в миллисекунды.
4. Используя аппаратное прерывания часов реального времени и режим будильника реализовать функции программируемого будильника.

**2. Листинг программы**

#include <dos.h>

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <io.h>

#include <windows.h>

unsigned int delayTime = 0;

unsigned int delayMs;

unsigned int date[6];

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

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

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

void getTime();

void setTime();

void delay(unsigned int);

void setAlarm();

void inputTime();

unsigned int bcdToDec(unsigned int);

unsigned int decToBcd(unsigned int);

void pByte(unsigned int byte);

void interrupt newDelay(...)

{

delayTime++;

outp(0x70, 0x0C);

inp(0x71);

outp(0x20, 0x20);

outp(0xA0, 0x20);

if (delayTime == delayMs)

{

puts("Delay's end");

disable();

setvect(0x70, oldDelay);

enable();

outp(0x70, 0x0B);

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

}

}

void interrupt newAlarm(...)

{

puts("Alarm!!!");

outp(0x70, 0x0C);

inp(0x71);

outp(0x20, 0x20);

outp(0xA0, 0x20);

//oldAlarm();

disable();

setvect(0x70, oldAlarm);

enable();

outp(0x70, 0x0B);

outp(0x71, inp(0x71) & 0xDF); // DF -- 11011111

}

int main()

{

while (1) {

printf("1 - Time\n");

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

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

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

printf("0 - Exit\n\n");

switch (getch()) {

case '1':

getTime();

break;

case '2':

setTime();

break;

case '3':

setAlarm();

break;

case '4':

fflush(stdin);

printf("Input delay (ms): ");

scanf("%u", &delayMs);

delay(delayMs);

break;

case '0':

printf("\n\n");

return 0;

default:

printf("\n\n");

break;

}

}

}

void pByte(unsigned int byte)

{

unsigned char bit;

for (int i = 7; i >= 0; i--)

{

bit = byte & 1;

byte /= 2;

printf("%c", bit + '0');

}

printf("\n");

}

void getTime()

{

unsigned char state;

printf("\n\n");

int i = 0;

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

{

outp(0x70, 0x0A);

state = inp(0x71);

if (state >> 7)

{

i--;

continue;

}

outp(0x70, dateReg[i]);

date[i] = inp(0x71);

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

}

printf("Date:\n%02u.%02u.%02u", date[2], date[1], date[0]);

printf(" %02u.%02u.20%02u\n\n", date[3], date[4], date[5]);

}

void setTime()

{

inputTime();

disable();

do { outp(0x70, 0x0A); } while (inp(0x71) >> 7);

outp(0x70, 0x0B);

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

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

{

outp(0x70, dateReg[i]);

outp(0x71, date[i]);

}

outp(0x70, 0x0B);

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

enable();

printf("\n\n");

}

void inputTime()

{

int n;

do {

fflush(stdin);

printf("Hours: ");

} while ((scanf("%d", &n) != 1 || n > 23 || n < 0));

date[2] = decToBcd(n);

do {

fflush(stdin);

printf("Minutes: ");

} while (scanf("%d", &n) != 1 || n > 59 || n < 0);

date[1] = decToBcd(n);

do {

fflush(stdin);

printf("Seconds: ");

} while (scanf("%d", &n) != 1 || n > 59 || n < 0);

date[0] = decToBcd(n);

}

void delay(unsigned int ms)

{

delayTime = 0;

disable();

oldDelay = getvect(0x70);

setvect(0x70, newDelay);

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

enable();

outp(0x70, 0x0B);

outp(0x71, inp(0x71) | 0x40); // 0x40 -- 01000000

return;

}

void setAlarm()

{

unsigned int alarmReg[] = { 0x01, 0x03, 0x05 };

inputTime();

disable();

oldAlarm = getvect(0x70);

setvect(0x70, newAlarm);

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

do { outp(0x70, 0x0A); } while (inp(0x71) >> 7);

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

{

outp(0x70, alarmReg[i]);

outp(0x71, date[i]);

}

enable();

outp(0x70, 0x0B);

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

printf("Alarm enable\n\n");

}

unsigned int bcdToDec(unsigned int bcd)

{

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

}

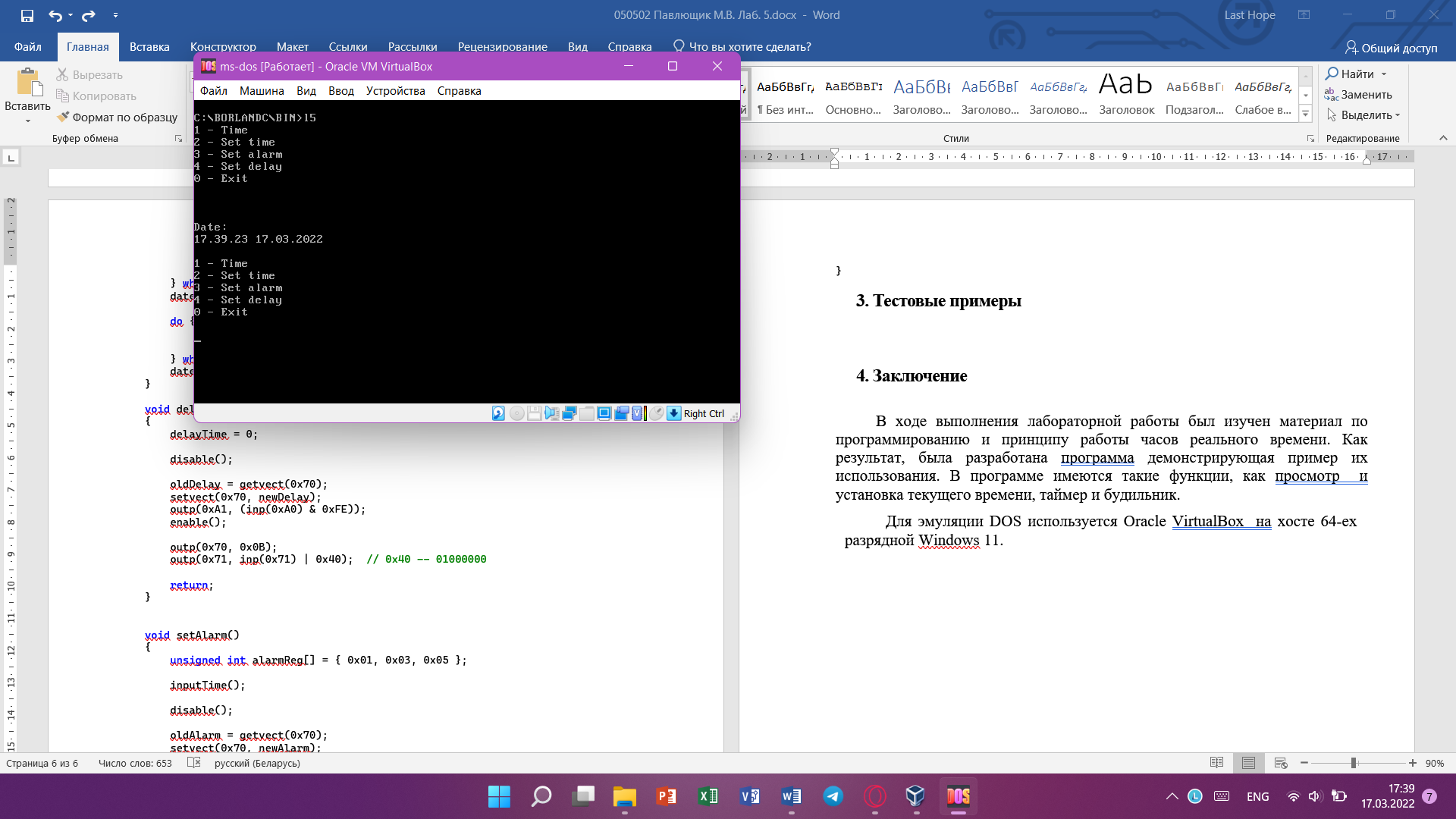
unsigned int decToBcd(unsigned int dec)

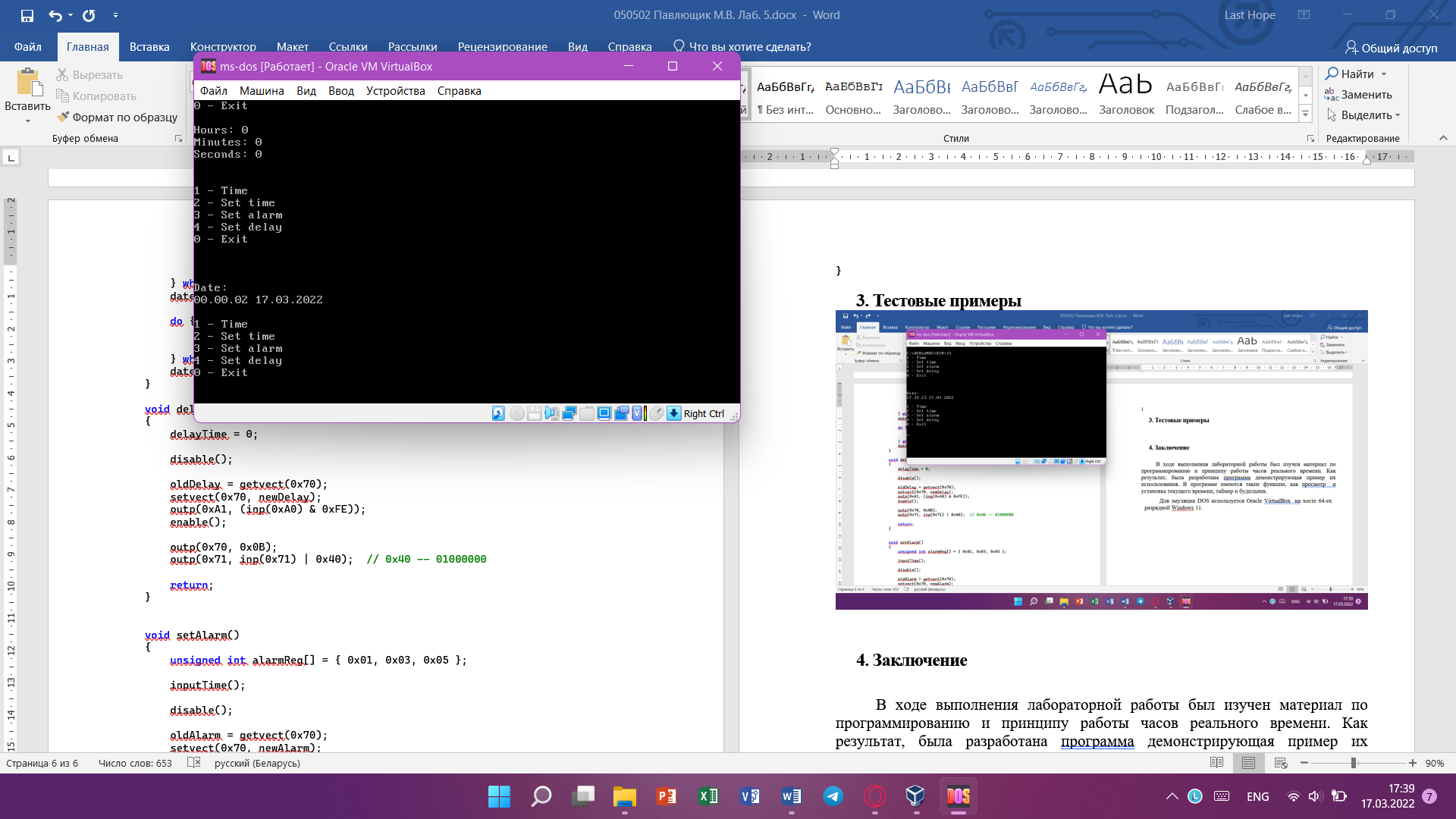
{

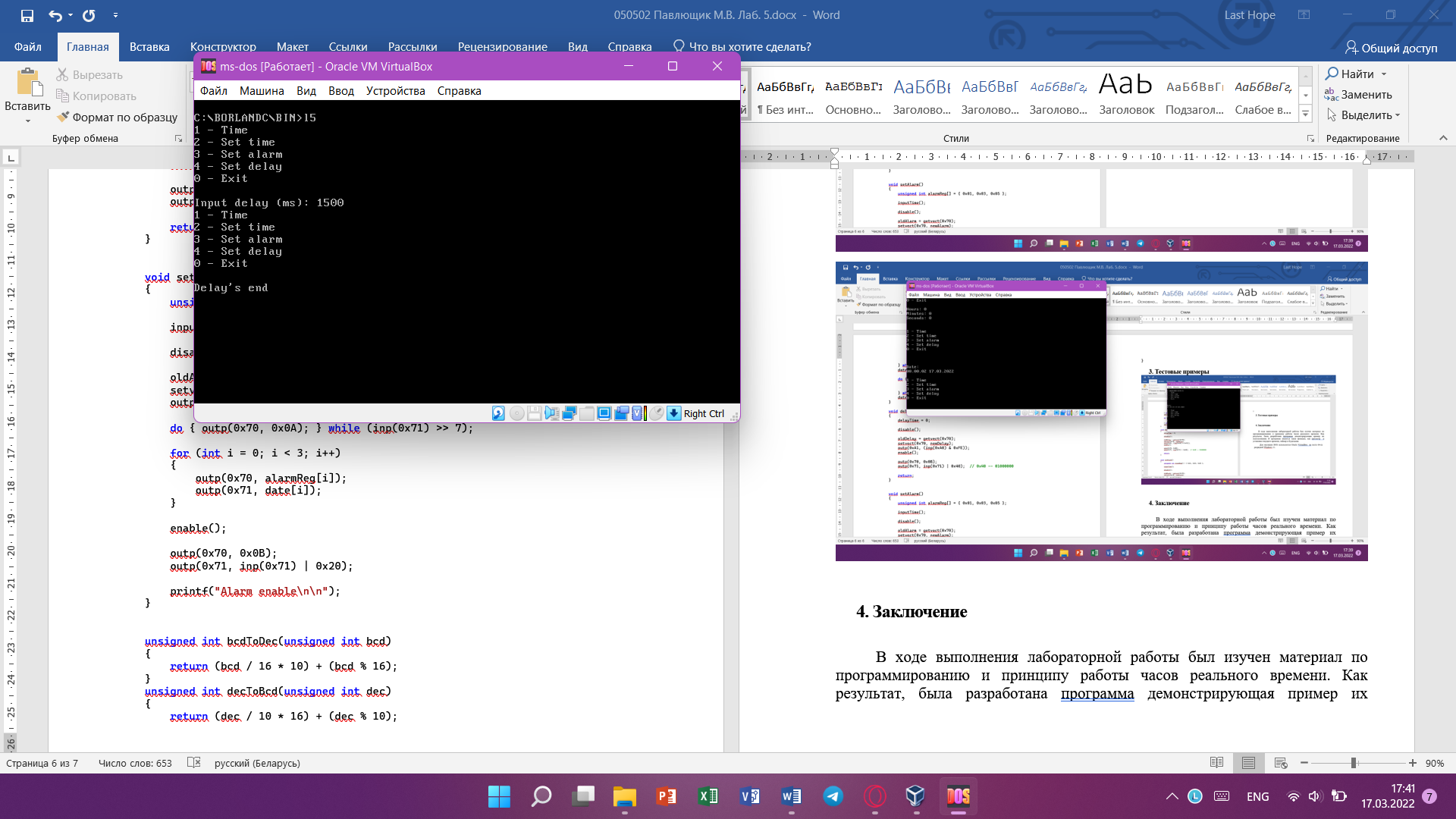
return (dec / 10 \* 16) + (dec % 10);

}

**3. Тестовые примеры**







**4. Заключение**

В ходе выполнения лабораторной работы был изучен материал по программированию и принципу работы часов реального времени. Как результат, была разработана программа демонстрирующая пример их использования. В программе имеются такие функции, как просмотр и установка текущего времени, таймер и будильник.

Для эмуляции DOS используется Oracle VM VirtualBox на хосте 64-ех разрядной Windows 11.