Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчёт по лабораторной работе №4

# «Программирование часов реального времени»

Проверил: Выполнил:

к.т.н., доцент студент гр., 250502

Одинец Дмитрий Николаевич Бригадир А. С.

Минск 2024

**Задание**

1. Написать программу, которая будет считывать и устанавливать время в часах реального времени. Считанное время должно выводиться на экран в удобочитаемой форме.

2.Используя аппаратное прерывание часов реального времени и режим генерации периодических прерываний реализовать функцию задержки с точностью в миллисекунды.

3. Используя аппаратное прерывания часов реального времени и режим будильника ЧРВ реализовать функции программируемого будильника.

**Листинг программы**

#include <stdio.h>

#include<stdlib.h>

#include<dos.h>

#include<string.h>

void showtime();

int delaytime = 0;

int kol = 0;

char\* reques[] = { "input hour","input minute" ,"input second" ,"input day" ,"input month" ,"input year" };

void interrupt(\*oldint70)(...);//прототипы старого прерывания

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

int registr[] = { 0x04,0x02,0x00,0x07,0x08,0x09 };//время и дата

char ms[3];

unsigned char todec(int val)//перевод в десятичное из BCD

{

return (unsigned char)((val / 16 \* 10) + (val % 16));

}

unsigned char tobcd(int val)//перевод в BCD

{

return (unsigned char)((val / 10 \* 16) + (val % 10));

}

void interrupt newint70alarm(...)//новое прерывание

{

outp(0x70, 0x0C);//команда для чтения состояния

if (inp(0x71) & 0x20)//5 бит прерывания будильника(1-есть,0-нет)

{

puts("work alarm\n");

}

disable();

setvect(0x70, oldint70);

enable();

outp(0x70, 0x0B);//команда регистра состояния

outp(0x71, (inp(0x71) & 0xDF));//выключаем 5 бит(запрещаем прерывание будильника)

}

void setalarm()

{

int value;

for (int k = 0; k < 3; k++)//пользователь вводит время будильника

{

printf("%s ", reques[k]);

scanf("%d", &value);

ms[k] = tobcd(value);

}

disable();

oldint70 = getvect(0x70);

setvect(0x70, newint70alarm);//устанавливаем новое прерывание

outp(0xA1, (inp(0xA0) & 0xFE)); // 0xFE - 1111 1110

// 0-й бит в 0 для разрешения прерывания от ЧРВ

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

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

{

outp(0x70, alarm[j]);//заносим в соответствующие регистры значения будильн.

outp(0x71, ms[j]);

}

enable();

outp(0x70, 0x0B);

outp(0x71, inp(0x71) | 0x20);//включаем 0010 0000 бит разрешения прерывания

printf("enabled in ");

for (k = 0; k < 3; k++)//выводим время будильника для информации

{

value = todec(ms[k]);

printf("%02d:", value);

}

printf("\n");

}

void interrupt newdelay(...)

{

printf("%d",kol);

kol++;

delaytime++;

//включаем 0010 0000 бит разрешения прерывания

outp(0x70, 0x0C); // выбор адреса в памяти CMOS (запись)

inp(0x71);

//// посыл сигнала контроллерам прерываний об окончании прерывания

outp(0x20, 0x20);

outp(0xA0, 0x20);

}

void setdelay()

{

printf("input delay in ms");

int delayuser;

scanf("%d", &delayuser);//пользователь вводит значение задержки

disable();

olddelay = getvect(0x70);

setvect(0x70, newdelay);

enable();

outp(0xA1, inp(0xA1) & 0xFE); // 0xFE = 1111 1110

// 0-й бит в 0 для разрешения прерывания от ЧРВ

outp(0x70, 0xB);//регистр состояния B

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

// 0x40 = 0100 0000

Периодическое прерывание вырабатывается с интервалом примерно 1 миллисекунда при

//установленном в 1 бите 6 регистра состояний B.

while (delaytime < delayuser);//прерывание срабатывает каждую секунду

//.в прерывании число delaytime увеличивается на один.то есть каждую секунду

// delaytime увеличивается на один и когда delaytime сравняется с delauser

// то мы переключим на стандартное прерывание

setvect(0x70, olddelay);

return;

}

void settime()

{

disable();

int a;

outp(0x70, 0x0A);//записываем команду что проверяем состояние1

do

{

a=inp(0x71) & 0x80;//проверяем 8 бит

} while (a == 1);

outp(0x70, 0x0B);//2 регистр состояния

outp(0x71, inp(0x71) | 0x80);//1000 0000или (с единицей переключает в единицу, с нулем оставляет значение которое было)

int neww;

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

{

printf("%s", reques[i]);

scanf("%d", &neww);

neww = tobcd(neww);

outp(0x70, registr[i]);

outp(0x71, neww);

}

outp(0x70, 0x0B);//2 регистр состояния

outp(0x71, inp(0x71) & 0x7F);//0111 1111

enable();

printf("new time\n");

showtime();

}

void showtime()

{

int i = 0;

for ( i = 0; i < 3; i++)//выводим время

{

outp(0x70, registr[i]);

printf("%02d:", todec(inp(0x71)));

}

printf("\n");

for (i = 3; i < 6; i++)//выводим дату

{

outp(0x70, registr[i]);

printf("%02d.", todec(inp(0x71)));

}

}

int main()

{

while (1)

{

printf("\n1-set time\n2-show time\n3-set alarm\n4 -set delay\ne-exit\n");

switch (getch())

{

case '1':

system("cls");

settime();

break;

case '2':

system("cls");

showtime();

break;

case'3':

system("cls");

setalarm();

break;

case'4':

system("cls");

setdelay();

break;

case'e':

return 0;

default:

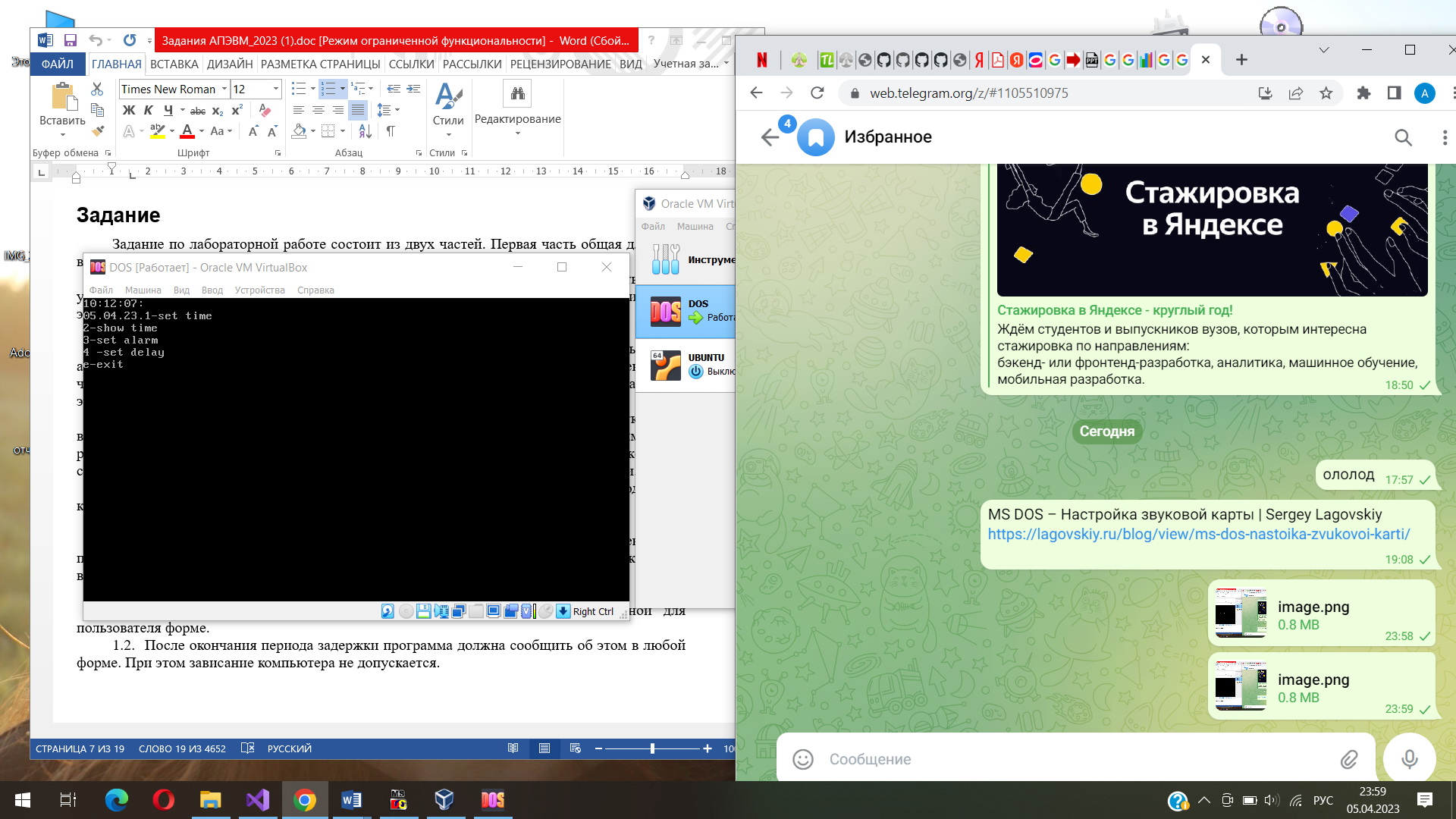
printf("wrong options");

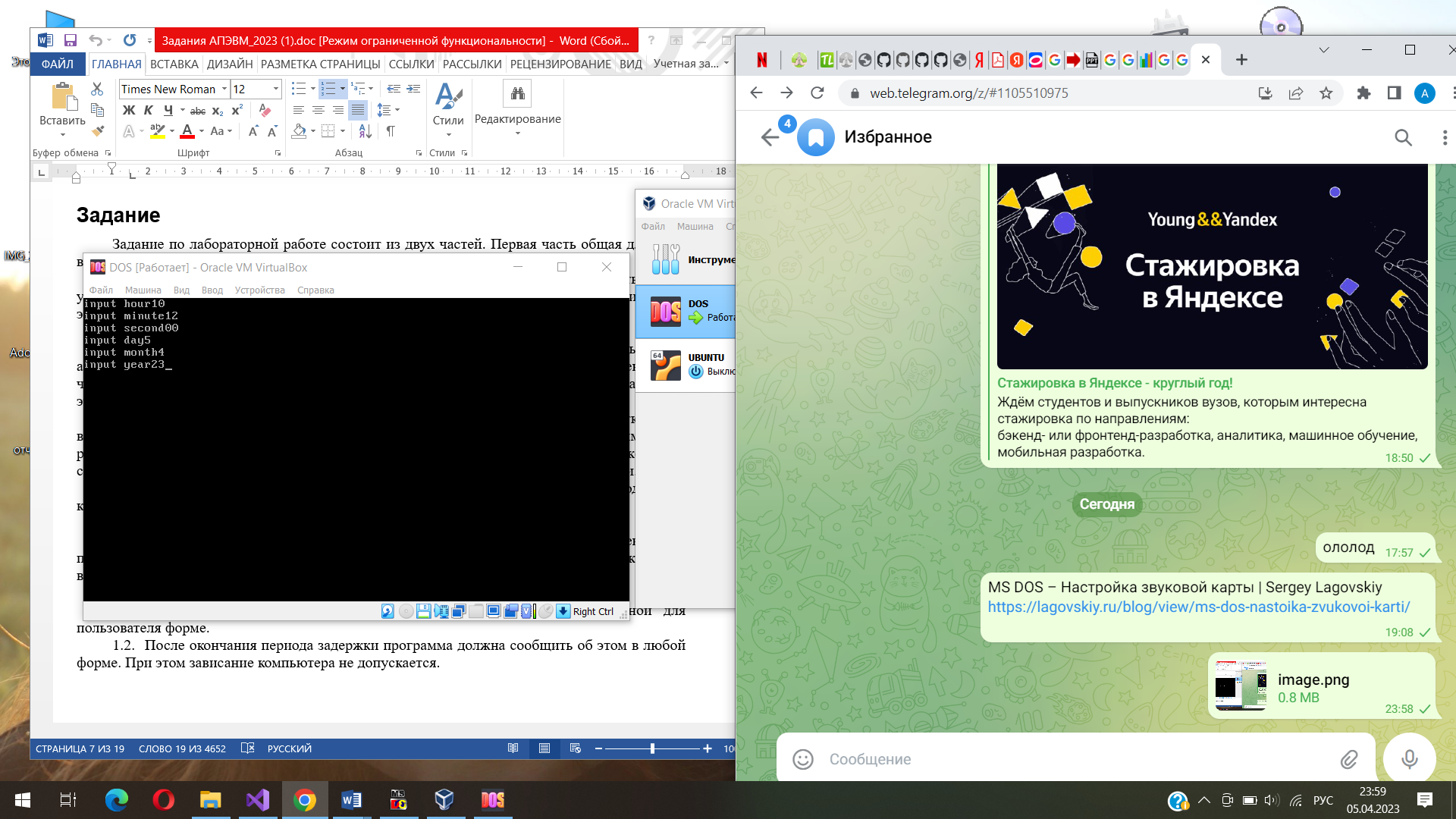
break;

}

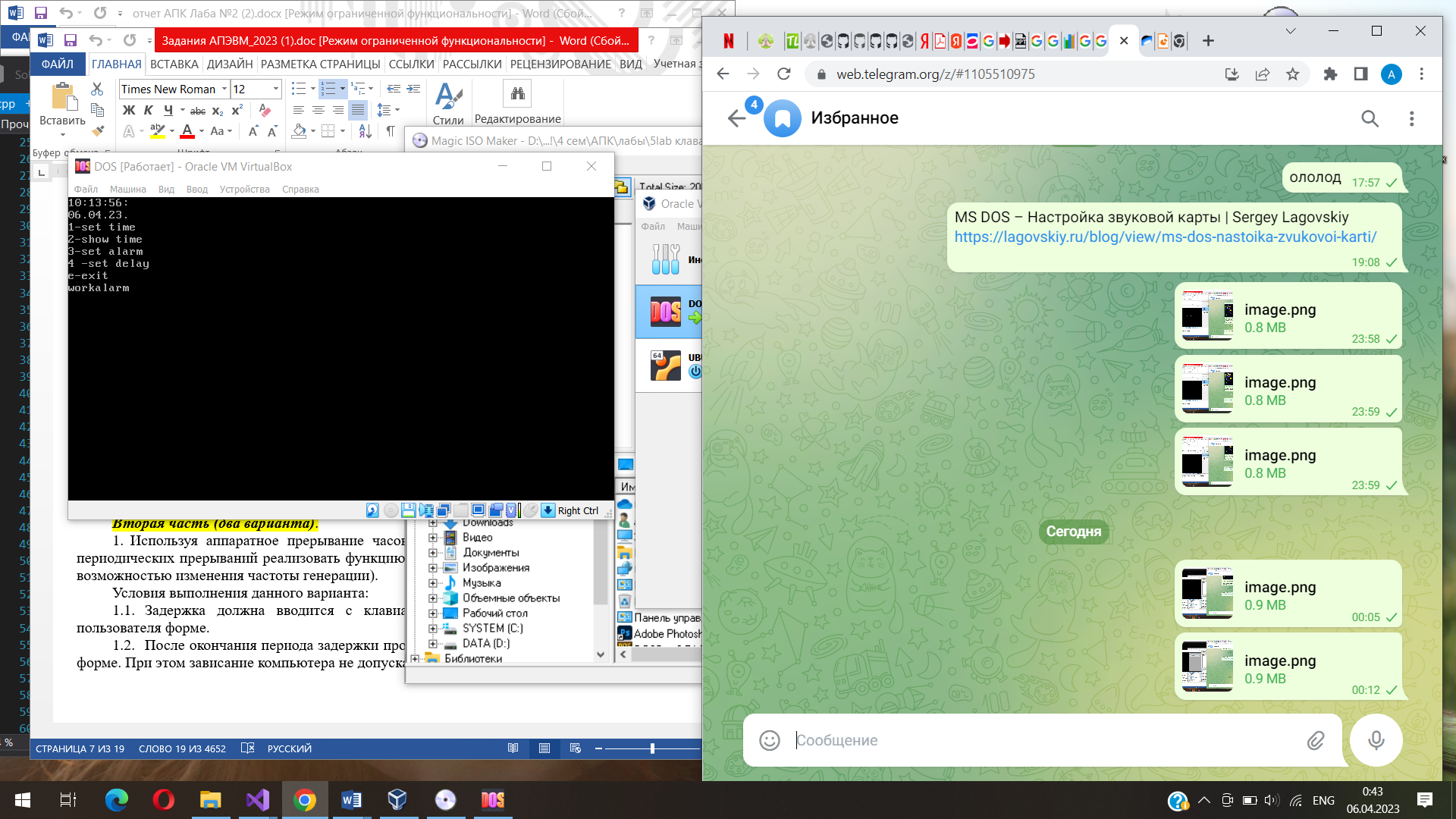
}

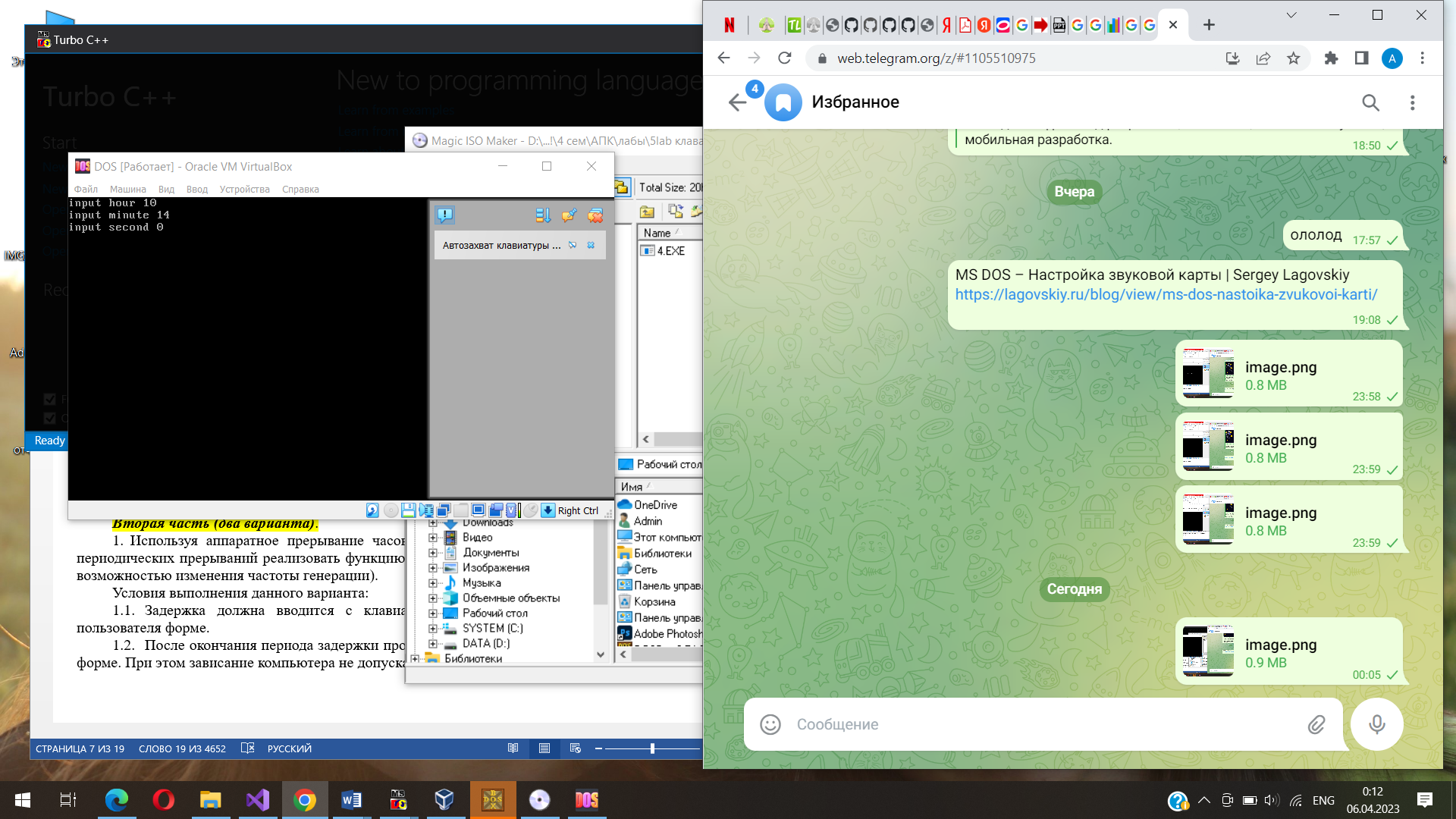
**Тест**

} 

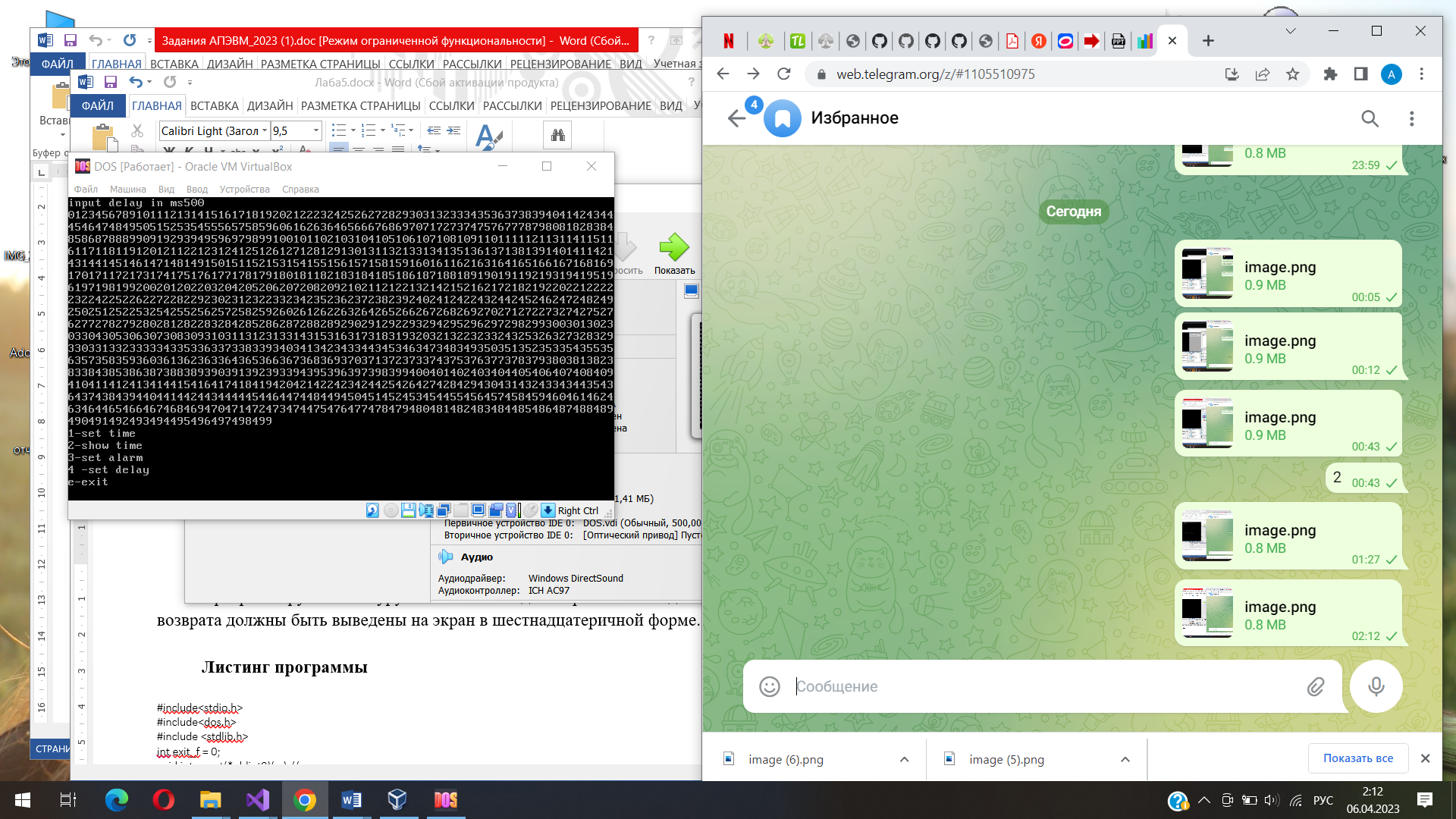
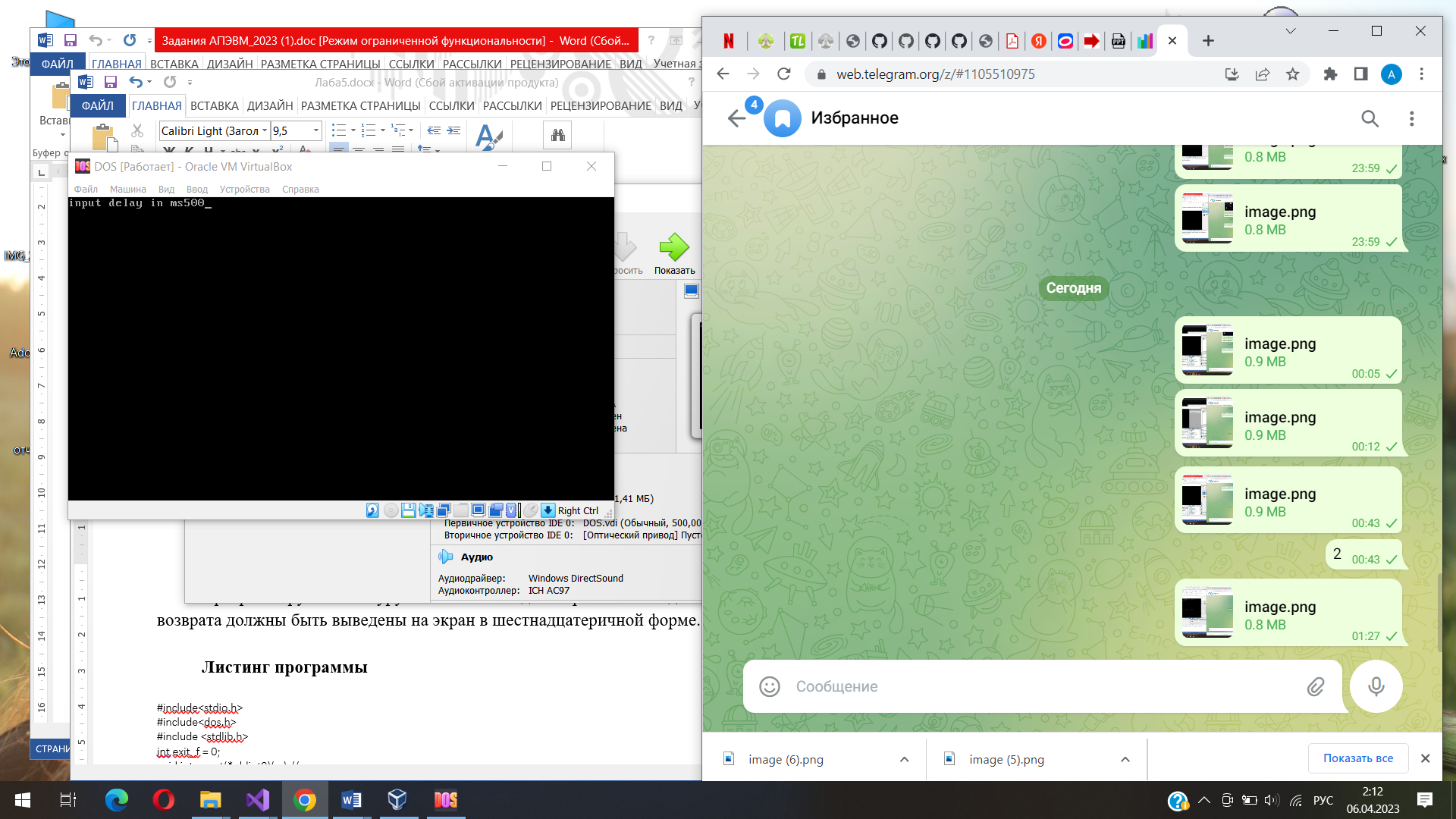


Установка времени Вывод времени





установка будильника Работа будильника



Установка задержки работа задержки