Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчёт по лабораторной работе №1

“Последовательный порт”

Проверил: Выполнил:

к.т.н., доцент студент гр.150502

Одинец Дмитрий Николаевич Альхимович Н. Г.

Минск 2023

**Задача**

Целью данной работы является разработка программного модуля реализации процедуры передачи (приема) байта информации через последовательный интерфейс.

Программа должна демонстрировать программное взаимодействие с последовательным интерфейсом с использованием следующих механизмов:

1. прямое взаимодействие с портами ввода-вывода (write, read)
2. использование BIOS прерывания 14h,
3. работа с COM-портом через регистры как с устройством ввода-вывода.

**Алгоритм**

Программа состоит из частей, представляющих собой некоторые функции. К ним относятся функции:

* инициализация COM-порта;
* запись данных в порт;
* чтение данных из порта;
* вывод результата на экран.

**Листинг программы**

Программа, взаимодействующая с портами ввода-вывода:

#include <windows.h>

#include <iostream>

using namespace std;

HANDLE port1, port2;

int main()

{

char buffer = 'c';

DWORD num\_of\_written, num\_of\_read;

port1 = ::CreateFile(L"COM1", GENERIC\_WRITE, 0, 0, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0); //function opens the COM-port

port2 = ::CreateFile(L"COM2", GENERIC\_READ, 0, 0, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0);

if(!port1)

cout << "Opening the COM-port 1 is impossible" << endl;

DCB properties = {0}; //DCB-structure describes main properties of the COM-port

properties.DCBlength = sizeof(properties);

if(!GetCommState(port1, &properties)) //function read DCB structure

cout << "Error" << endl;

properties.BaudRate = CBR\_9600;

properties.ByteSize = 8;

properties.StopBits = ONESTOPBIT;

properties.Parity = NOPARITY;

if(!SetCommState(port2, &properties)) //function configures the COM-port

cout << "Configuring the COM-port is impossible" << endl;

WriteFile(port1, &buffer, sizeof(buffer), &num\_of\_written, NULL);

cout << sizeof(buffer) << "/" << num\_of\_written << " bytes written." << endl;

buffer = ' ';

ReadFile(port2, &buffer, num\_of\_written, &num\_of\_read, 0);

if(num\_of\_read > 0)

cout << "Read data: " << buffer << endl;

else cout << "The buffer is empty" << endl;

CloseHandle(port1);

CloseHandle(port2);

return 0;

}

Программа, использующая BIOS прерывание 14h:

.model small

.stack 100h

.data

error db "Nothing was read$"

message db "Read data: $"

.code

mov ax, @data

mov ds, ax

initialise:

mov ah, 00h ;function initialises the COM-port

mov al, 11100011b ;111 - 9600 (speed), 00 - no parity, 0 - 1 stopbit, 11 - 8 bits

int 14h

write:

mov al, 'c'

mov ah, 01h ;function writes a symbol to COM-port

mov dx, 0 ;the first COM-port (n=1, n-1=0)

int 14h

read:

mov ah, 02h ;function reads a symbol from COM-port

mov dx, 1 ;the second COM-port

int 14h

output:

cmp al, 0

je not\_read

lea dx, message

mov ah, 09h

int 21h

mov ah, 02h

xor dx, dx

mov dl, al

int 21h

exit:

mov ax, 4c00h

int 21h

ret

not\_read:

lea dx, error

mov ah, 09h

int 21h

Программа, работающая с COM-портом через регистры как с устройством ввода-вывод:

.model small

.stack 100h

.data

error db "Nothing was read$"

buffer db ?

message db "Read data: $"

.code

mov ax, @data

mov ds, ax

initialise:

mov al, 80h ;7th bit - 1 (Divisor Latch Access Bit) => 3F8h and 3F9h - speed

mov dx, 3FBh ;Line Control Register

out dx, al

mov dx, 3F8h ;controlling register, in charge of transmitting data. "low" byte

mov al, 00h ;0 - port is used for writing data

out dx, al

mov al, 0Ch ;"high" byte, 0Ch - 12 - 9600 (speed)

mov dx, 3F9h ;register controlls interruptions

out dx, al

mov dx, 3FCh ;port controls the modem

mov al, 00001011b ;DTR, RTS, OUT1, OUT2

out dx, al

mov dx, 3F9h

mov al, 0

out dx, al

write: ;using COM1

xor al, al

mov dx, 3FDh

in al, dx

mov dx, 3F8h

mov al, 'c'

out dx, al

read: ;using COM2

xor al, al

mov dx, 3FDh ;ready for reading data

in al, dx

mov dx, 3F8h

in al, dx

mov buffer, al

output:

cmp buffer, 0

je not\_read

lea dx, message

mov ah, 09h

int 21h

mov ah, 02h

xor dx, dx

mov dl, buffer

int 21h

exit:

mov ax, 4C00h

int 21h

ret

not\_read:

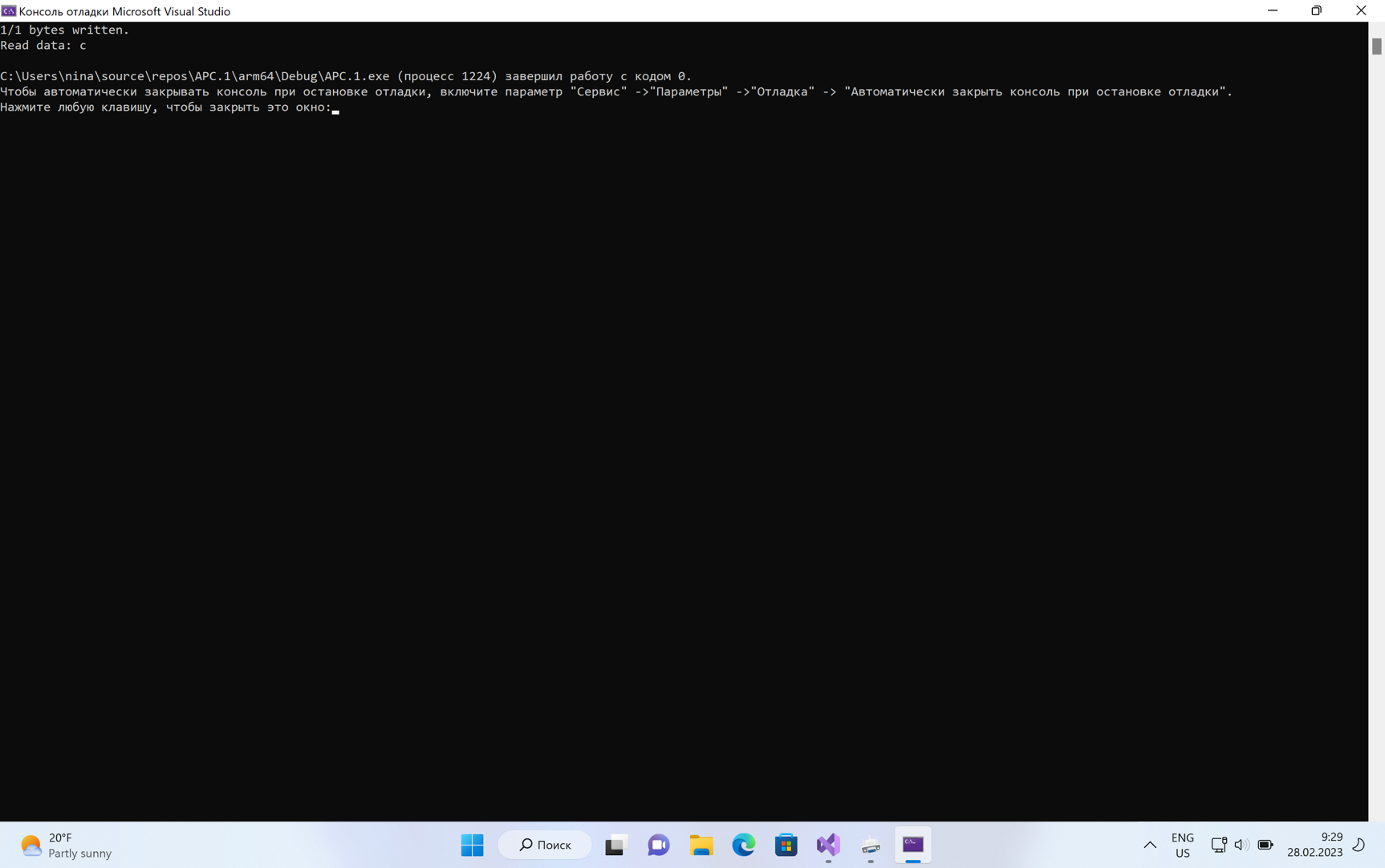
lea dx, error

mov ah, 09h

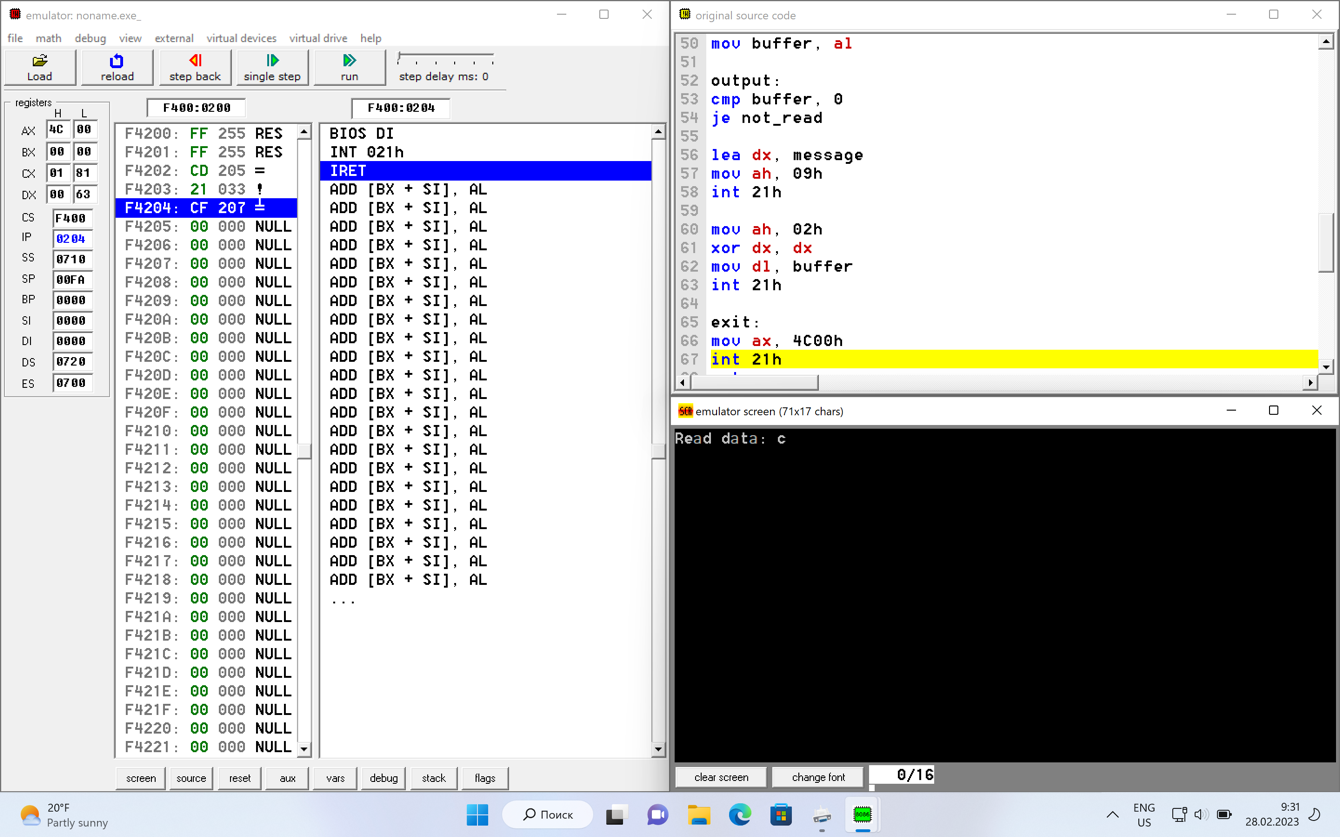
int 21h

**Тест**

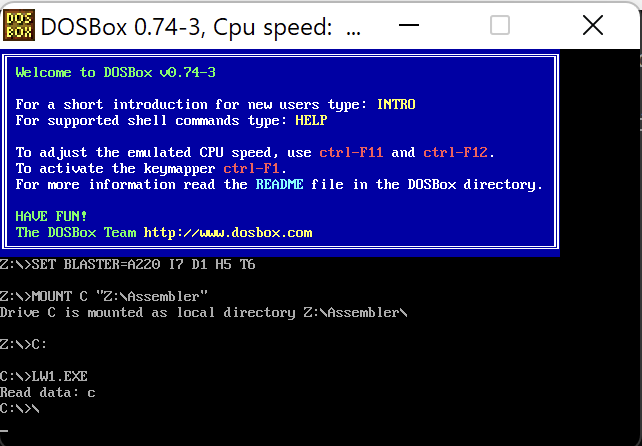
Программа, взаимодействующая с портами ввода-вывода:

****

Программа, использующая BIOS прерывание 14h:

****

Программа, работающая с COM-портом через регистры как с устройством ввода-вывод:

****

**Заключение**

В ходе данной лабораторной был разработан программный модуль реализации процедуры передачи (приема) байта информации через последовательный интерфейс с использованием различных механизмов.

Для эмуляции COM-портов использовался Virtual Serial Port Driver.