

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе №1

«Последовательный порт»

Выполнил:
студент гр. 150503
Ефимчик А. И.

Проверил:
к.т.н, донецнт
Одинец Д. Н.

Минск 2023

1 ПОСТАНОВКА ЗАДАЧИ

Разработать программный модуль реализации процедуры передачи (приёма) 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 устройствами ввода-вывода.

2 АЛГОРИТМ

Программа состоит из нескольких подпрограмм (частей программы), представляющих собой некоторые функции. К ним относятся функции:

- Инициализация порта
- Запись байта информации в порт
- Чтение байта информации из порта
- Вывод результата на экран

3 ЛИСТИНГ ПРОГРАММЫ

Далее приведены листинги программ, реализующие различные механизмы передачи (приёма) информации через последовательный интерфейс.

3.1 Листинг программы, взаимодействующей с портами ввода-вывода

```
#include <windows.h>
#include <iostream>
#include <sstream>

#include "logger.hpp"

HANDLE COM_Port_1;
LPCTSTR Port_Name_1 = "COM3";
HANDLE COM_Port_2;
LPCTSTR Port_Name_2 = "COM4";
```

```

void Read_from_COM()
{
    DWORD Size;
    char Received_Char;
    ReadFile(COM_Port_2, &Received_Char, 1, &Size, nullptr);
    if (Size>0) {
        std::cout << "Received char: " << Received_Char << std::endl;
    }
}

int main()
{
    COM_Port_1 = CreateFile(Port_Name_1, GENERIC_WRITE, 0,
        nullptr, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, nullptr);
    COM_Port_2 = CreateFile(Port_Name_2, GENERIC_READ, 0,
        nullptr, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, nullptr);

    if (COM_Port_1==INVALID_HANDLE_VALUE || COM_Port_2==INVALID_HANDLE_VAL
    if (GetLastError()==ERROR_FILE_NOT_FOUND) {
        LOG_ERROR("COM-port does not exist!");
        return 1;
    }
    LOG_ERROR("Unspecified error");
    return 1;
}

DCB Serial_Params = { 0 };
Serial_Params.DCBlength = sizeof(Serial_Params);
if (!GetCommState(COM_Port_1, &Serial_Params)) {
    LOG_ERROR("get COM state error");
    return 1;
}
Serial_Params.BaudRate = CBR_9600;
Serial_Params.ByteSize = 8;
Serial_Params.StopBits = ONESTOPBIT;

```

```

Serial_Params.Parity = NOPARITY;
if (!SetCommState(COM_Port_2, &Serial_Params)) {
LOG_ERROR("set COM state error");
return 1;
}

char data = 'h';
DWORD Size = sizeof(data);
DWORD Bytes_Written;

if (WriteFile(COM_Port_1, &data, Size, &Bytes_Written,
  nullptr)==0) {
LOG_ERROR("Write COM failed");
return 1;
}

std::stringstream ss;
ss << Size << " bytes exists, " << Bytes_Written << "
  bytes sent ";
LOG_INFO(ss.str());

Read_from_COM();

CloseHandle(COM_Port_1);
CloseHandle(COM_Port_2);
LOG_INFO("Handlers closed");

return 0;
}

```

3.2 Листинг программы, использующей BIOS прерывание 14h

```

.model small
.stack 100h

.data

```

```
Error_Write_Message db "Write error!",0Dh,0Ah,'$'  
Error_Read_Message db "Read error!",0Dh,0Ah,'$'
```

```
.code
```

```
jmp start
```

```
Init_COM1 proc
```

```
    xor ax,ax  
    mov al,10100011b  
    mov dx,0  
    int 14h  
    ret
```

```
Init_COM1 endp
```

```
Write_COM1 proc
```

```
    mov al,'A' ; symbol  
    mov ah,1   ; write func  
    mov dx,0   ; port num  
    int 14h    ; COM interrupt  
    test al,80h  
    jnz NoWrite  
    ret
```

```
Write_COM1 endp
```

```
NoWrite proc
```

```
    mov ah,9  
    mov dx,offset Error_Write_Message  
    add dx,2  
    int 21h  
    ret
```

```
NoWrite endp
```

```
Read_COM2 proc
```

```
    mov ah,2 ; func num (read with wait)
```

```

        mov dx,1      ; port num (COM2)
        int 14h       ; COM interrupt
        test al,80h   ; 1000 0000 if no time-out
        jnz NoRead
        ret
Read_COM2 endp

NoRead proc
        mov ah,9
        mov dx,offset Error_Read_Message
        add dx,2
        int 21h
        ret
NoRead endp

Output proc
        mov ah,02h   ; print func
        mov dl,al
        int 21h
        ret
Output endp

Exit proc
        mov ax,4C00h
        int 21h
        ret
Exit endp

start:
        call Init_COM1
        call Write_COM1
        call Read_COM2

        call Output
        call Exit

```

```
end start
```

3.3 Листинг программы, работающей с СОМ-портами через регистры как с устройствами ввода-вывода

```
.model small
.stack 100h

.data

Error_Write db "Write error!",0Dh,0Ah,'$'
Error_Read db "Read error!",0Dh,0Ah,'$'
Information db "Byte sent: $"
Data_Byte db 'A'
Data_Byte2 db ?

.code

Init_COM1 proc
    mov al,80h
    mov dx,3FBh
    out dx,al

    mov dx,3F8h
    mov al,00h
    out dx,al
    mov al,0Ch
    mov dx,3F9h
    out dx,al

    mov dx,3FCh
    mov al,00001011b
    out dx,al

    mov dx,3F9h
    mov al,0
    out dx,al
```

```

        ret
Init_COM1 endp

IsWrite_COM1 proc
    xor al,al
    mov dx,3FDh
    in al,dx
    test al,10h
    jnz NoWrite
    ret
IsWrite_COM1 endp

NoWrite proc
    mov ah,9
    mov dx,offset Error_Write
    int 21h
    ret
NoWrite endp

IsRead_COM2 proc
    xor al,al
    mov dx,3FDh
    in al,dx
    test al,10b
    jnz NoRead
    ret
IsRead_COM2 endp

NoRead proc
    mov ah,9
    mov dx,offset Error_Read
    int 21h
    ret
NoRead endp

Send_Byte proc

```



```

        mov dx,3F8h
        mov al,Data_Byte
        out dx,al
        ret
Send_Byte endp

Read_Byte proc
        mov dx,3F8h
        in al,dx
        mov Data_Byte2,al
        ret
Read_Byte endp

Exit proc
        mov ax,4C00h
        int 21h
        ret
Exit endp

start:
        mov ax,@data
        mov ds,ax
        call Init_COM1
        call IsWrite_COM1
        call Send_Byte
        mov al,2
        call IsRead_COM2
        call Read_Byte
        mov dx,offset Information
        mov ah,9
        int 21h
        mov ah,02h
        mov dl,Data_Byte2
        int 21h
        call Exit

```

end start

4 РЕЗУЛЬТАТ РАБОТЫ

```
Y:\LAB_01\cmake-build-debug>LAB_01.exe  
[INFO] - 2023-02-14 07:52:43 - main.cpp:63 - 1 bytes exists, 1 bytes sent  
Received char: h  
[INFO] - 2023-02-14 07:52:43 - main.cpp:69 - Handlers closed
```

Рисунок 4.1 – Результат передачи и приема байта с использованием windows.h на C++

```
DOSBox version 0.74-3  
Copyright 2002-2019 DOSBox Team,  
---  
CONFIG:Loading primary settings  
MIDI:Opened device:win32  
Serial1: Opening COM3  
Serial2: Opening COM4  
Failed to create hardware surfac  
Restarting video subsystem with  
Drive Y is mounted as lo  
Z:\>SET PATH=Z:\;Y:\B086  
Z:\>Y:  
Y:\>cd LAB_01  
Y:\LAB_01>LAB1_2.EXE  
A  
Y:\LAB_01>
```

Рисунок 4.2 – Результат передачи и приема байта с использованием прерывания 21h на TASM

```
---  
CONFIG:Loading primary settings  
MIDI:Opened device:win32  
Serial1: Opening COM3  
Serial2: Opening COM4  
Failed to create hardware surfac  
Restarting video subsystem with  
Copyright (C) Microsoft C  
Run File [LAB1_03.EXE]:  
List File [NUL.MAP]:  
Libraries [.LIB]:  
Y:\LAB_01>LAB1_03.EXE  
Byte sent: A  
Y:\LAB_01>
```

Рисунок 4.3 – Результат передачи и приема байта с использованием регистров на TASM

5 ЗАКЛЮЧЕНИЕ

В ходе лабораторной удалось передать 1 байт информации через последовательный порт с использованием различных механизмов.