Белорусский государственный университет информатики и радиоэлектроники

# Кафедра ЭВМ

### Отчет

### по лабораторной работе № 5

### «Детерминированные методы доступа к моноканалу»

Проверил:

Марцинкевич В.А.

Выполнил:

студент группы 130501

Гнездилов А.М.

Минск 2022

**Цель работы:** написать программу пакетной передачи данных через COM-порты упрощенным алгоритмом Token Ring в соответствии с требованиями.

**Листинг кода**

**Main.cpp**

#include "USER.h"

void f(list \*list)

{

USER\* temp = list->GetHead();

while (1)

{

temp->marker = \*list->GetMarker();

Sleep(5000);

temp->marker = 0;

temp = temp->next;

}

}

int menu()

{

system("cls");

cout << "1 - Добавить пользователя" << endl

<< "2 - Список пользователей" << endl

<< "3 - Передать данные" << endl

<< "4 - Получить данные" << endl

<< "0 - Выход" << endl;

return \_getch();

}

int main()

{

system("chcp 1251 > null");

HANDLE handle1 = ::CreateFile(L"COM1", GENERIC\_WRITE, 0, 0, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0);

HANDLE handle2 = ::CreateFile(L"COM2", GENERIC\_READ, 0, 0, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0);

list list;

int i = 0;

while (1)

{

switch (menu())

{

case '1': { list.add(); i++; } break;

case '2': list.GetList(); break;

case '3': list.SendData(handle1, handle2); break;

case '4': list.GetInfo(); break;

case '0': { system("cls"); cout << "Работа завершена."; return 0; }

}

if (i == 1)

{

thread thr(f, &list);

thr.detach();

}

}

return 0; }

**USER.h**

#include <windows.h>

#include <iostream>

#include "conio.h"

#include <vector>

#include <stdlib.h>

#include <stack>

#include <cmath>

#include <thread>

using namespace std;

struct USER

{

string name;

string dataFrom;

string nameFrom;

string nameTo;

string dataTo;

USER\* next;

int marker = 0;

};

class list

{

private:

USER\* head, \* tail;

int marker;

public:

USER\* GetHead();

list();

int\* GetMarker();

void SetMarker(int m);

void add();

void GetList();

void SendData(HANDLE handle1, HANDLE handle2);

void GetInfo();

};

**USER.cpp**

#include "USER.h"

struct str

{

list\* list;

USER\* temp;

HANDLE handle1, handle2;

};

USER\* list::GetHead()

{

return head;

}

string code(string data)

{

string ret;

for (int i = 0; i < data.length(); i++)

{

stack<int> ST;

int count = 0;

string str;

int a, t, d;

t = 0;

d = 1;

a = data[i];

while (a)

{

t += (a % 2) \* d;

a = a / 2;

d = d \* 10;

}

do

{

ST.push(t % 10);

t /= 10;

count++;

} while (t);

count = 8 - count;

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

str += '0';

while (!ST.empty())

{

t = ST.top();

ST.pop();

char s = { char(t + '0') };

str += s;

}

ret += str;

}

return ret;

}

string decode(string data)

{

string ret;

string str;

int s = 8;

for (int i = 0; i < data.length() / s; i++)

{

for (int j = i \* s; j < i \* s + s; j++)

{

str += data[j];

}

int st = 0;

int ch = 0;

for (int j = str.length() - 1; j >= 0; j--)

{

if (str[j] == '1') ch += pow(2, st);

st++;

}

char a = ch;

ret += a;

str.clear();

}

return ret;

}

void f2(str S)

{

while (true)

{

if (S.temp->marker == 2)

{

DWORD size = 0;

char buff;

string data;

int fl = 0;

do

{

ReadFile(S.handle2, &buff, 1, &size, NULL);

if (buff != '~')

data += buff;

} while (buff != '~');

PurgeComm(S.handle2, PURGE\_RXABORT | PURGE\_TXABORT | PURGE\_RXCLEAR | PURGE\_TXCLEAR);

string str;

string sour;

string pour;

int sum;

for (int i = 0; i < 10; i++)

if (data[i] != '|')

pour += data[i];

for (int i = 10; i < 20; i++)

if (data[i] != '|')

sour += data[i];

for (int i = 20; i < data.length() - 1; i++)

str += data[i];

sum = data[data.length() - 1];

string info = decode(str);

string d;

d += info;

d += " [";

d += str;

d += "]";

S.temp->dataFrom = d;

S.temp->nameFrom = sour;

S.list->SetMarker(1);

return;

}

}

}

void f1(str S)

{

while (true)

{

if (S.temp->marker == 1)

{

S.list->SetMarker(2);

char data1[256] = "";

char data2[256] = "";

string NameTo = S.temp->nameTo, NameFrom = S.temp->name;

string ADR = S.temp->nameTo;

string str;

string b;

for (int i = 0; i < S.temp->dataTo.length(); i++)

data1[i] = S.temp->dataTo[i];

system("cls");

char s = strlen(data1);

if (NameTo.length() < 10)

for (int i = NameTo.length(); i < 10; i++)

NameTo += '|';

if (NameFrom.length() < 10)

for (int i = NameFrom.length(); i < 10; i++)

NameFrom += '|';

str += NameTo;

str += NameFrom;

b = code(data1);

str += b;

str += s;

str += '~';

for (int i = 0; i < str.length(); i++)

data2[i] = str[i];

DWORD size = strlen(data2);

DWORD bytes;

BOOL Ret = WriteFile(S.handle1, &data2, size, &bytes, NULL);

while (S.temp->name != ADR)

S.temp = S.temp->next;

if (S.temp->name == ADR)

{

thread thr(f2, S);

thr.detach();

}

return;

}

}

}

list::list()

{

head = tail = NULL;

marker = 1;

}

int\* list::GetMarker()

{

return &marker;

}

void list::SetMarker(int m)

{

marker = m;

}

void list::add()

{

system("cls");

USER\* temp = new USER;

cout << "Введите имя пользователя: ";

cin >> temp->name;

if (head != NULL)

{

tail->next = temp;

tail = temp;

tail->next = head;

}

if (head == NULL)

{

head = tail = temp;

tail->next = head;

}

system("cls");

}

void list::GetList()

{

system("cls");

USER\* temp = head;

int i = 1;

if (temp != NULL)

do

{

cout << i++ << " - " << temp->name << endl;

temp = temp->next;

} while (temp != head);

else cout << "Список пуст!" << endl;

\_getch();

}

void list::SendData(HANDLE handle1, HANDLE handle2)

{

system("cls");

USER\* temp = head;

int i = 1, s, p;

if (temp != NULL)

{

do

{

cout << i++ << " - " << temp->name << endl;

temp = temp->next;

} while (temp != head);

cout << "\nКакой пользователь хочет передать данные?: ";

cin >> s;

cout << "Какому пользователю хочет передать данные?: ";

cin >> p;

temp = head;

for (int i = 0; i < p - 1; i++)

temp = temp->next;

string name = temp->name;

temp = head;

for (int i = 0; i < s - 1; i++)

temp = temp->next;

temp->nameTo = name;

system("cls");

cout << "Введите данные: ";

cin >> temp->dataTo;

str S;

S.list = this;

S.temp = temp;

S.handle1 = handle1;

S.handle2 = handle2;

thread thr(f1, S);

thr.detach();

}

else

{

cout << "Список пуст!" << endl;

\_getch();

}

}

void list::GetInfo()

{

while(true)

{

system("cls");

USER\* temp = head;

int i = 1;

do

{

cout << "Пользователь №" << i++ << endl

<< "Имя: " << temp->name << endl

<< "Последние полученные данные: " << temp->dataFrom << endl

<< "Имя отпавителя: " << temp->nameFrom << endl

<< "Состояние: " << temp->marker << endl

<< "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl << endl;

temp = temp->next;

} while (temp != head);

Sleep(3000);

if (\_kbhit()) return;

}

}

**Скриншоты работы программы**







