Министерство науки и образования Российской Федерации

Пензенский государственный университет

Кафедра «Системы автоматизированного проектирования»

**Отчет**

По лабораторной работе №5

По курсу «Операционные системы реального времени»

На тему «Работа с разделяемой памятью и файлами в асинхронном режиме»

Выполнил студент гр.20ВВС1

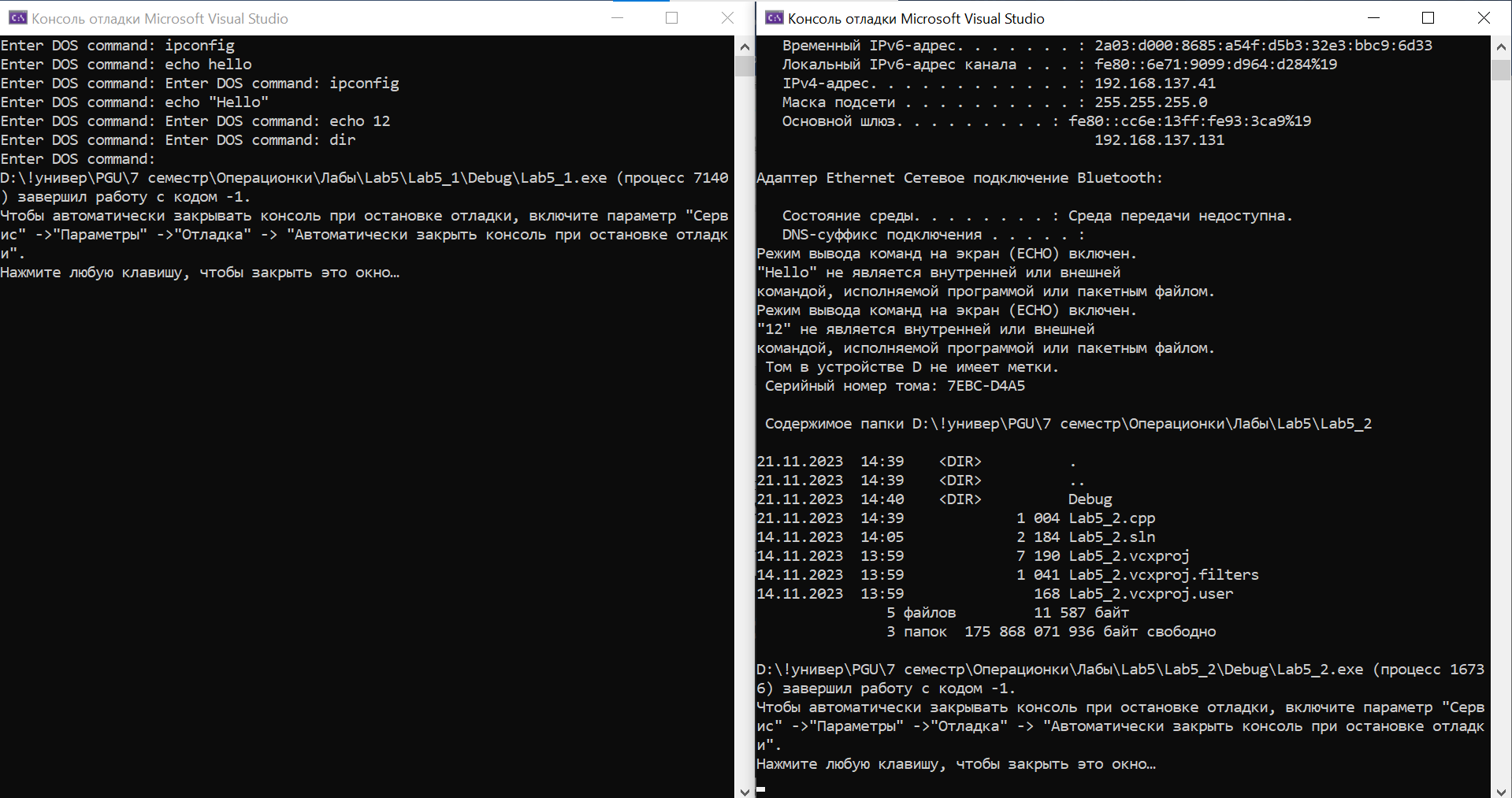
Горбунов Н.А.

Проверил:

Гудков П.А.

Пенза, 2023

**Задание 7:** разработать систему из двух программ. Первая программа предназначена для ввода команд DOS. Вторая программа демонстрирует результаты выполнения команд. Синхронизация с использованием семафоров.



**Листинг:**

**Программа для ввода DOS команд:**

#include <iostream>

#include <string>

#include <windows.h>

#include <process.h>

#include "synchapi.h"

#include <iostream>

#include <fstream>

#include <mutex>

int main()

{

HANDLE hSemaphore = CreateSemaphore(NULL, 1, 2, TEXT("MySemaphore"));

std::string command;

WaitForSingleObject(hSemaphore, INFINITE);

while (true) {

std::cout << "Enter DOS command: ";

std::cin >> command;

std::ofstream outfile("../../test.txt");

outfile << command << std::endl;

outfile.close();

ReleaseSemaphore(hSemaphore, 1, NULL);

WaitForSingleObject(hSemaphore, INFINITE);

//system(command.c\_str());

//std::cout << "\n\n\n";

}

CloseHandle(hSemaphore);

return 0;

}

//std::string command;

// while (true) {

// std::cout << "Enter DOS command: ";

// std::cin >> command;

// system(command.c\_str());

// std::cout << "\n\n\n";

// }

// return 0;

**Программа для вывода DOS команд:**

#include <iostream>

#include <windows.h>

#include <fstream>

#include <string>

using namespace std;

int main() {

HANDLE semaphore = OpenSemaphore(SEMAPHORE\_ALL\_ACCESS, false, TEXT("MySemaphore"));

if (semaphore == NULL) {

cout << "Error opening semaphore\n";

return 1;

}

while (true) {

WaitForSingleObject(semaphore, INFINITE); // Wait for the semaphore

string line;

ifstream ini\_file{ "../../test.txt" };

getline(ini\_file, line);

// Run DOS command

/\*PROCESS\_INFORMATION pi = { 0 };

STARTUPINFO si = { sizeof(PROCESS\_INFORMATION), 0, 0, 0, 0 };

CreateProcess(NULL, (LPWSTR)line.c\_str(), NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);

GetExitCodeProcess(pi.hProcess, &pi.dwProcessId);

CloseHandle(pi.hThread);\*/

system(line.c\_str());

//cout << "Result of command: " << line << endl;

ReleaseSemaphore(semaphore, 1, NULL);// Release the semaphore so that other processes can use it

}

CloseHandle(semaphore);

return 0;

}

**Вывод:** научился работать с семафорами между программами и синхронизироваться с их помощью.