

Министерство образования Республики Беларусь

Учреждение образования

Брестский государственный технический университет

Лабораторная работа №9

по теме: «Строки и манипуляции с файловыми системами»

Выполнил:

Студент 3-го курса

Группы АС-51

Сац А.В.

Проверила:

Давидюк Ю.И.

Брест 2020

Цель работы:

Закрепить навыки решения задач на строки и манипуляции с файловыми системами, научиться проектировать и разрабатывать динамические библиотеки в ОС Windows и Linux.

Задание – Windows

Разработать динамическую библиотеку DLL, включающую функцию (функции), реализующую следующий функционал для работы со строками (использовать только стандартные средства – библиотека string.h). Продемонстрировать ее подключение и использование.

- 1) Выполнить операцию «перекрещивания» двух строк, переданных в качестве параметров (для заданных позиций в обеих строках, первая часть новой первой строки берется из первой переданной строки до заданной позиции, а вторая часть – из второй переданной после заданной позиции во второй строке, для второй строки – тоже самое, только берутся альтернативные части оригинальных строк), полученные строки конкатенировать. Пример: для строк «abcd» и «hgjfs» и точек скрещивания 2 и 3 (для первой и второй строки соответственно) результат скрещивания будет «abfs» и «hgjcd», после итоговой конкатенации – «abfshgjcd».

Файлы DLL библиотеки:

Library.h

```
#pragma once
#ifndef LIBRARY_EXPORTS
#define LIBRARY_API __declspec(dllexport)
#else
#define LIBRARY_API __declspec(dllimport)
#endif

extern "C" LIBRARY_API std::string function(std::string str1_1, std::string str2_1, int dot1_1, int dot2_1);
```

Library.cpp

```
#include "pch.h" // use stdafx.h in Visual Studio 2017 and earlier
#include <iostream>
#include <string.h>
#include "Library.h"

std::string function(std::string str1_1, std::string str2_1, int dot1_1, int dot2_1) {
    std::string str1, str2, str3, str4;
    str1 = str1_1;
    str2 = str2_1;
    int dot1 = dot1_1, dot2 = dot2_1;
    for (int i = 0; i < dot1; i++)
        str3.push_back(str1[i]);
    for (int i = dot2; i < str2.length(); i++)
        str3.push_back(str2[i]);
    for (int i = 0; i < dot2; i++)
        str4.push_back(str2[i]);
    for (int i = dot1; i < str1.length(); i++)
        str4.push_back(str1[i]);
    str3 += str4;
    return str3;
}
```

Файлы клиентского приложения

Client.cpp

```
#include <iostream>
#include "Library.h"

using namespace std;

int main()
{
    setlocale(LC_ALL, "rus");
    string str1, str2;
    cout << "Введите строку 1: "; cin >> str1;
    cout << "Введите строку 2: "; cin >> str2;

    cout << endl;

    int dot1, dot2;
    cout << "Введите точку перекрещивания 1: "; cin >> dot1;
    cout << "Введите точку перекрещивания 1: "; cin >> dot2;

    cout << "\nРезультат конкатенации: " << function(str1, str2, dot1, dot2) <<
endl;
}
```

Результат работы программы



Консоль отладки Microsoft Visual Studio

```
Введите строку 1: abcd
Введите строку 2: hgjfs

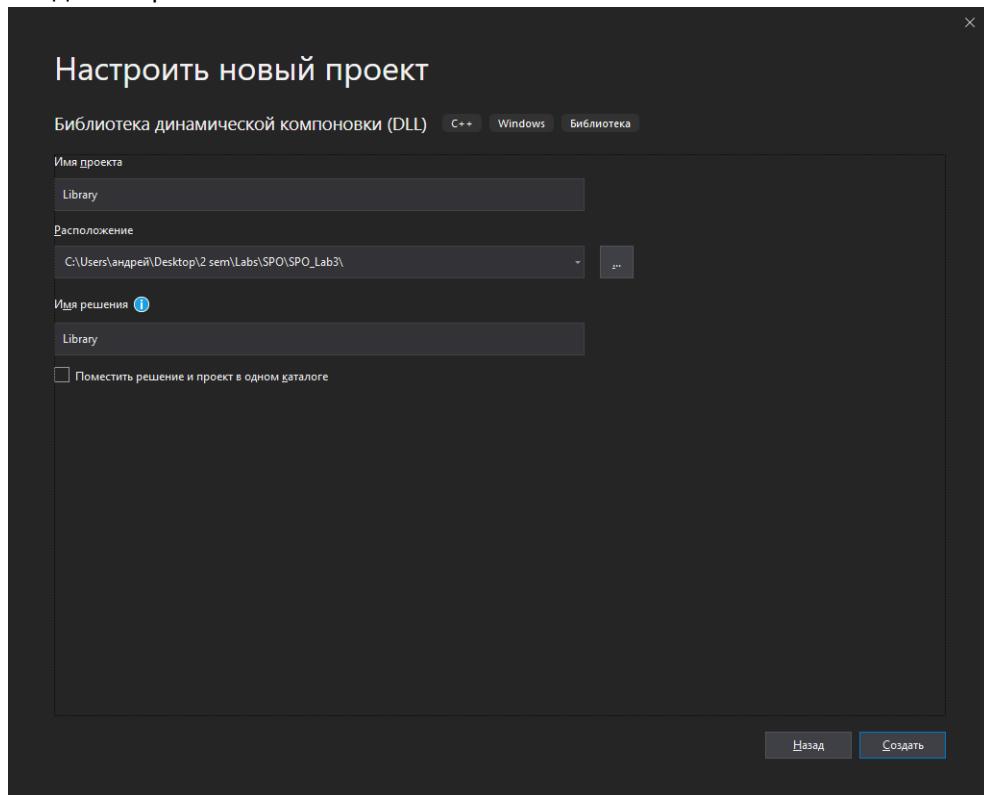
Введите точку перекрещивания 1: 2
Введите точку перекрещивания 1: 3

Результат конкатенации: abfshgjcd

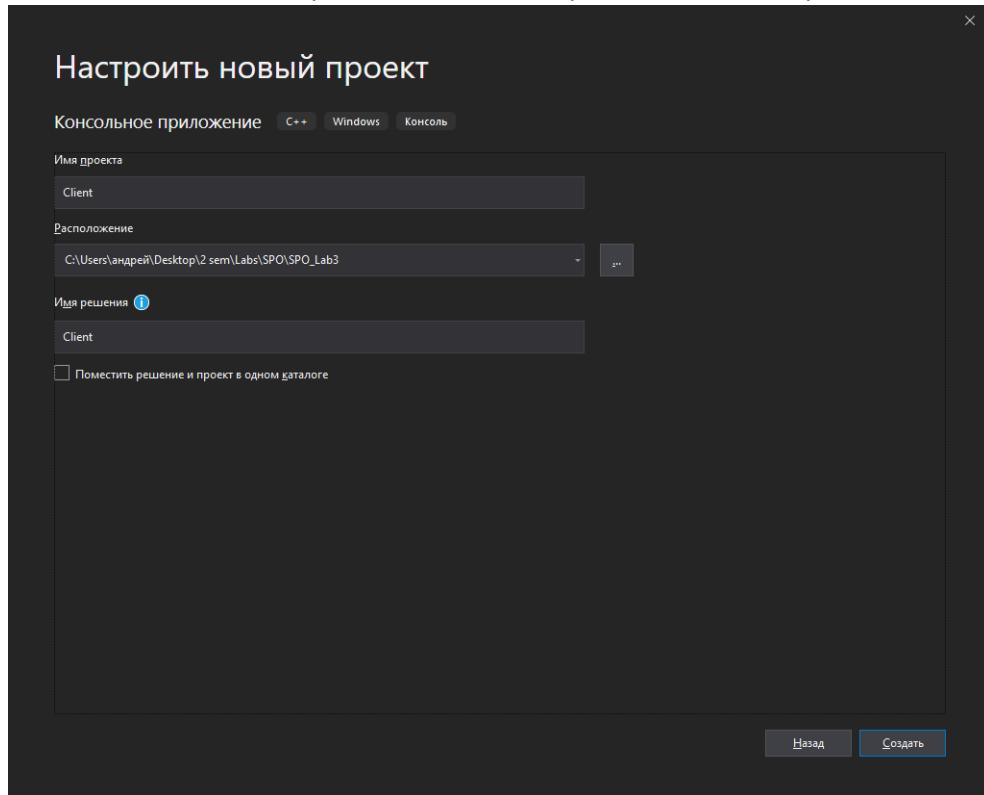
C:\Users\андрей\Desktop\2 sem\Labs\SPO\SPO_Lab3\Client\Debug\Client.exe (процесс 13400) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Инструкция по созданию и подключению DLL

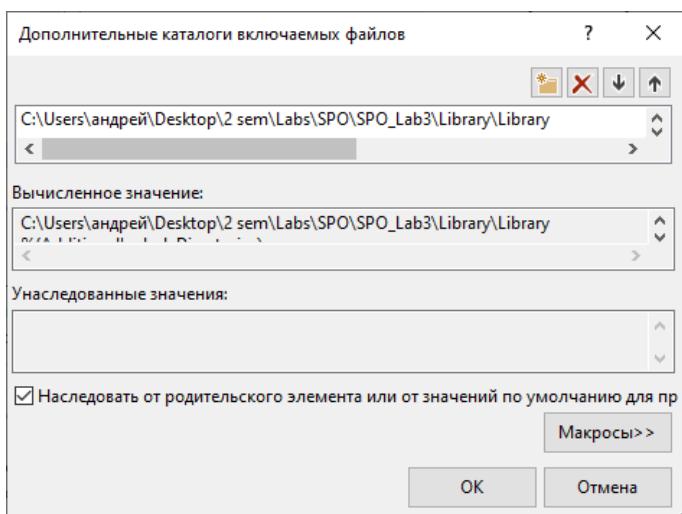
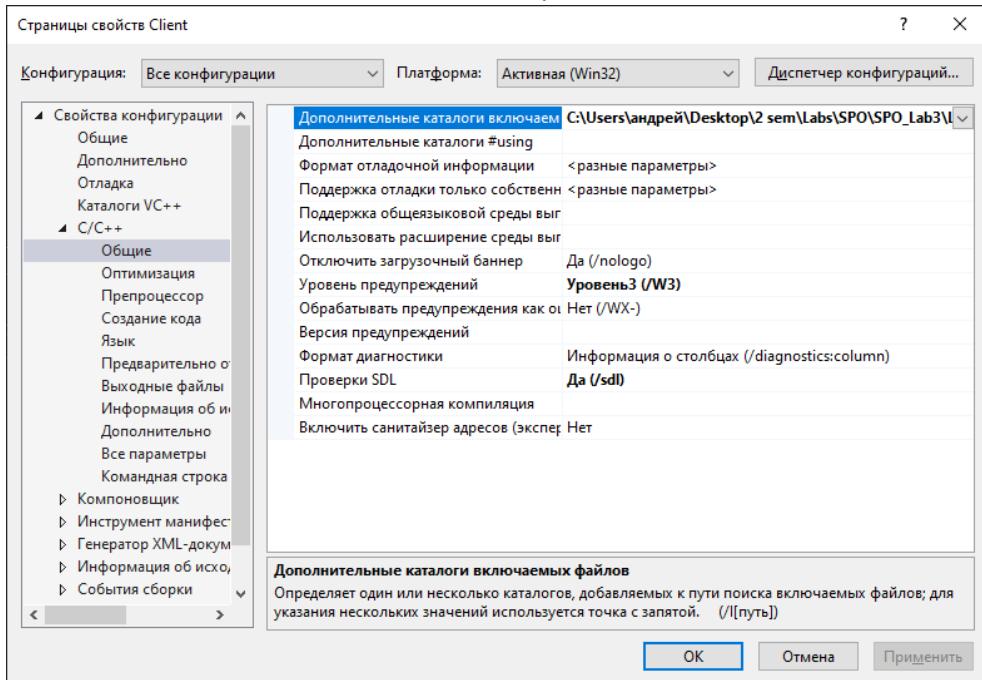
1. Создание проекта библиотеки DLL



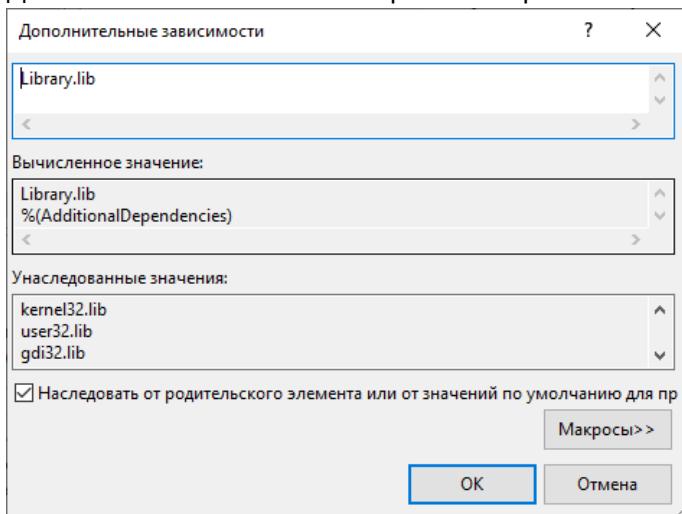
2. Создание клиентского приложения, использующего библиотеку DLL

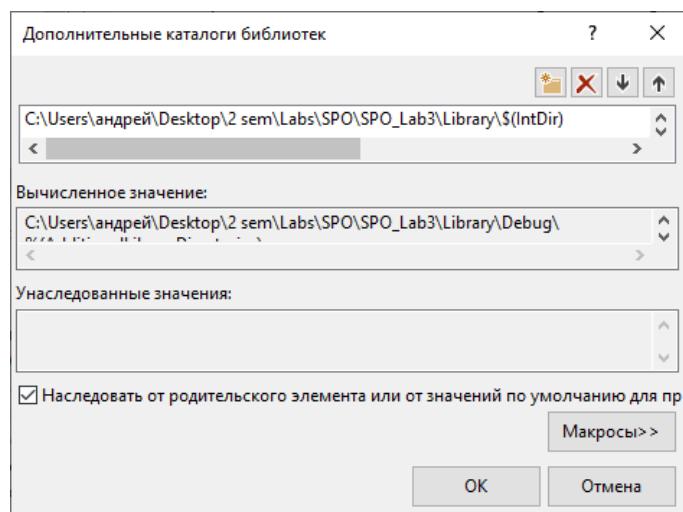
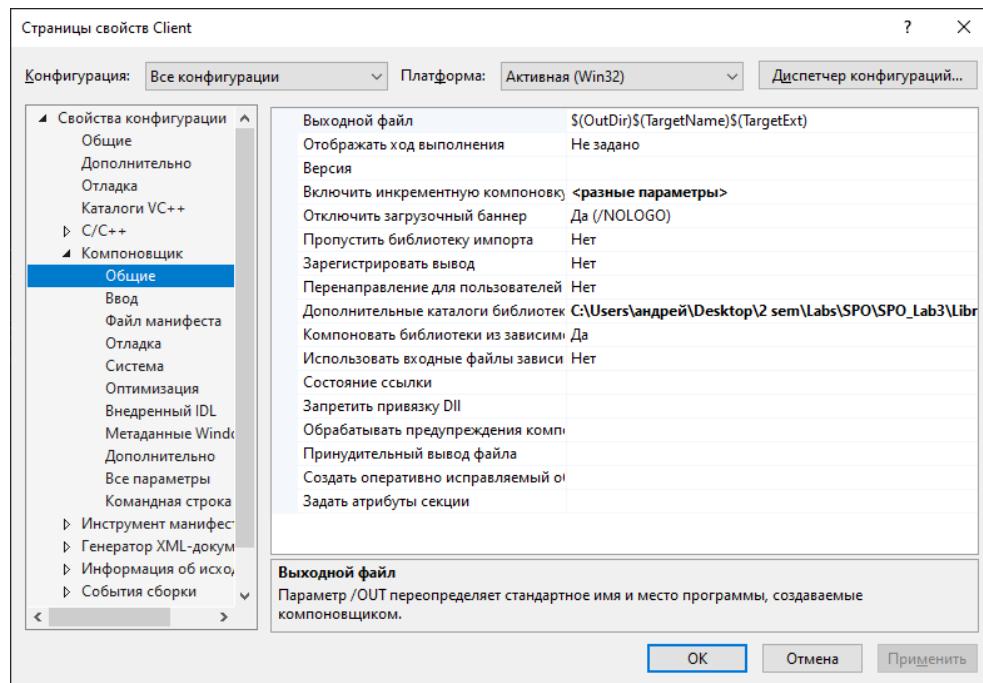


3. Добавление заголовка библиотеки DLL в путь включения

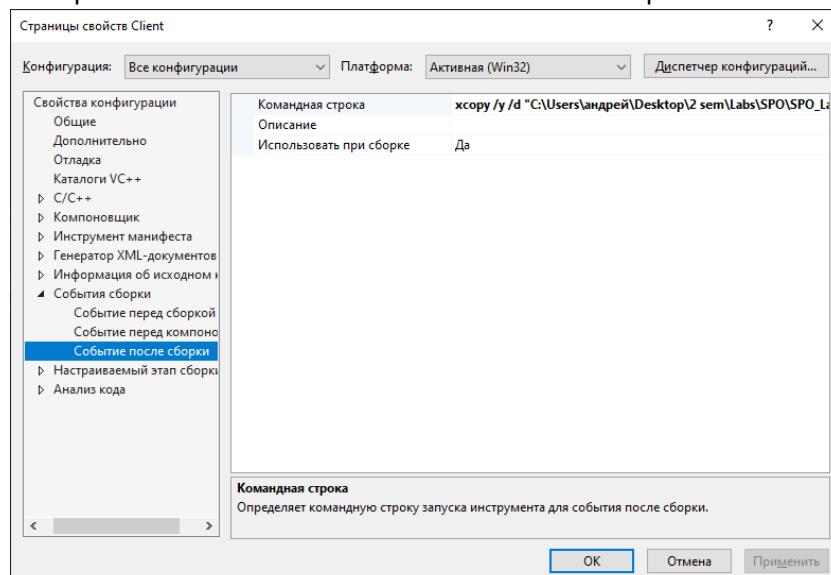


4. Добавление библиотеки импорта DLL в проект





5. Копирование библиотеки DLL в событие после сборки



Задания – Linux

Разработать динамическую библиотеку so, включающую функцию (функции), реализующую следующий функционал. Продемонстрировать ее подключение и использование. Для поиска необходимых функций использовать ресурс <https://www.die.net>.

- 3) Функция, читающая содержимое указанной директории и выводящая его на экран.

Содержимое библиотеки

Library.cpp

```
#include <stdio.h>
#include <sys/types.h>
#include <dirent.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>

void library(char* str){
    DIR *dir;
    struct dirent *entry;
    dir = opendir(str);
    if (!dir) {
        perror("diropen");
    }

    while ( (entry = readdir(dir)) != NULL) {
        printf("%s\n",
               entry->d_name);
    }

    closedir(dir);
}
```

Содержимое исполняемого файла

Main.cpp

```
#include <stdio.h>
#include <sys/types.h>
#include <dirent.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>

void library(char* str);

int main() {
    printf("Enter the adres: ");
    char str[50];
    scanf("%s", &str);
    library(str);
}
```

Скриншоты

```
ubuntu@ubuntu: ~/Lab9
File Edit View Search Terminal Help
ubuntu@ubuntu:~/Lab9$ g++ -fPIC -c Library.cpp
ubuntu@ubuntu:~/Lab9$ g++ -shared -o libdyn.so Library.o
g++: error: Laibrary.o: No such file or directory
g++: fatal error: no input files
compilation terminated.
ubuntu@ubuntu:~/Lab9$ g++ -shared -o libdyn.so Library.o
ubuntu@ubuntu:~/Lab9$ g++ -c Main.cpp
Main.cpp: In function ‘int main()’:
Main.cpp:13:18: warning: format ‘%s’ expects argument of type ‘char*’, but argument 2 has type ‘char (*)[50]’ [-Wformat=]
    scanf("%s", &str);
               ^
ubuntu@ubuntu:~/Lab9$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/library
ubuntu@ubuntu:~/Lab9$ g++ Main.o -L. -ldyn -o rez
ubuntu@ubuntu:~/Lab9$ ./res
bash: ./res: No such file or directory
ubuntu@ubuntu:~/Lab9$ ./rez file
Enter the adres: /home/ubuntu/Lab9
.
..
rez
Main.o
libdyn.so
Library.o
Main.cpp
Library.cpp
ubuntu@ubuntu:~/Lab9$
```

