

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
Куцун Ю. В.
Проверила:
Давидюк Ю.И.

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.

8. Для строки-параметра получить строку-результат конкатенацией n копий исходной строки.

Файлы DLL библиотеки:

Dll.h

```
#pragma once
#ifdef DLL_EXPORTS
#define DLL_API_declspec(dllexport) #else
#define DLL_API_declspec(dllimport) #endif

extern "C" DLL_API std::string fun(std::string str, int n1);
```

Dll.cpp

```
#include "pch.h"
#include <iostream>
#include <string.h>
#include "Dll.h"

std::string fun(std::string str, int n1){
    std::string str1=str, str2;
    int n=n1;
    for(int i=0;i<n;i++)
        str2+=str1;
    return str2;
}
```

Файлы клиентского приложения

main.cpp

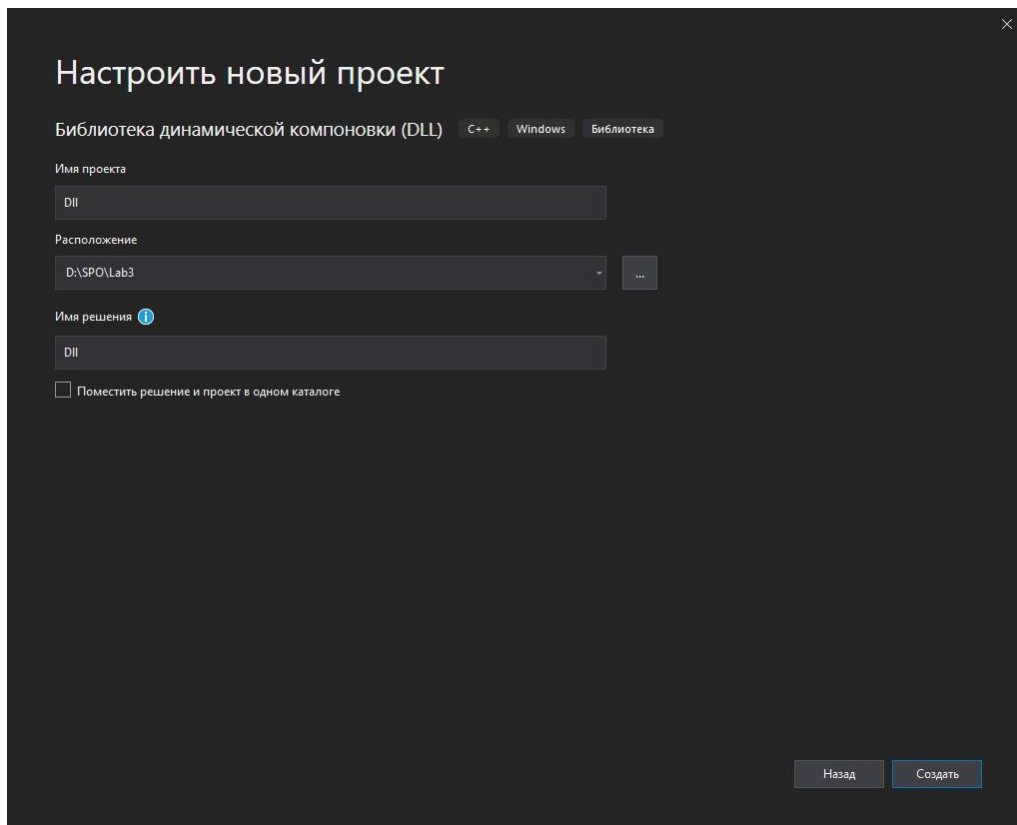
```
#include <iostream>
#include "Dll.h"

using namespace std;

int main() {
    string str;
    int n;
    cin>>str>>n;
    cout<<fun(str, n);
    return 0;
}
```

Инструкция по созданию и подключению DLL

1. Создание проекта библиотеки DLL



Настроить новый проект

Библиотека динамической компоновки (DLL) C++ Windows Библиотека

Имя проекта
Dll

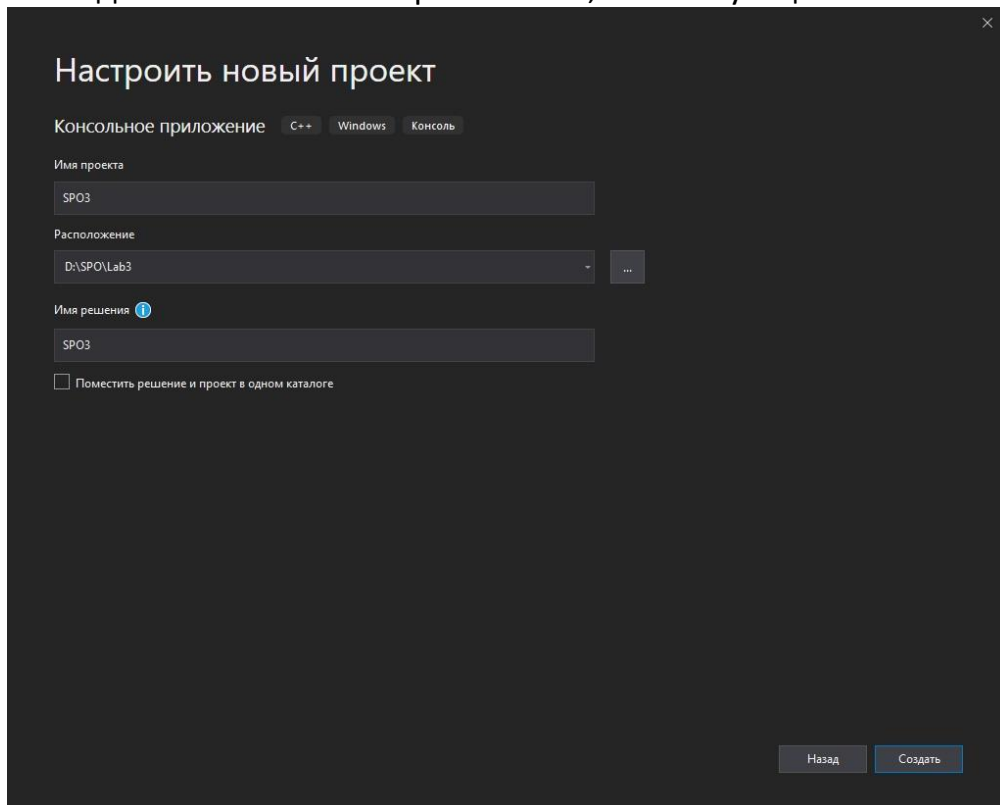
Расположение
D:\SPO\Lab3

Имя решения 1
Dll

☐ Поместить решение и проект в одном каталоге

Назад Создать

2. Создание клиентского приложения, использующего библиотеку DLL



Настроить новый проект

Консольное приложение C++ Windows Консоль

Имя проекта
SPO3

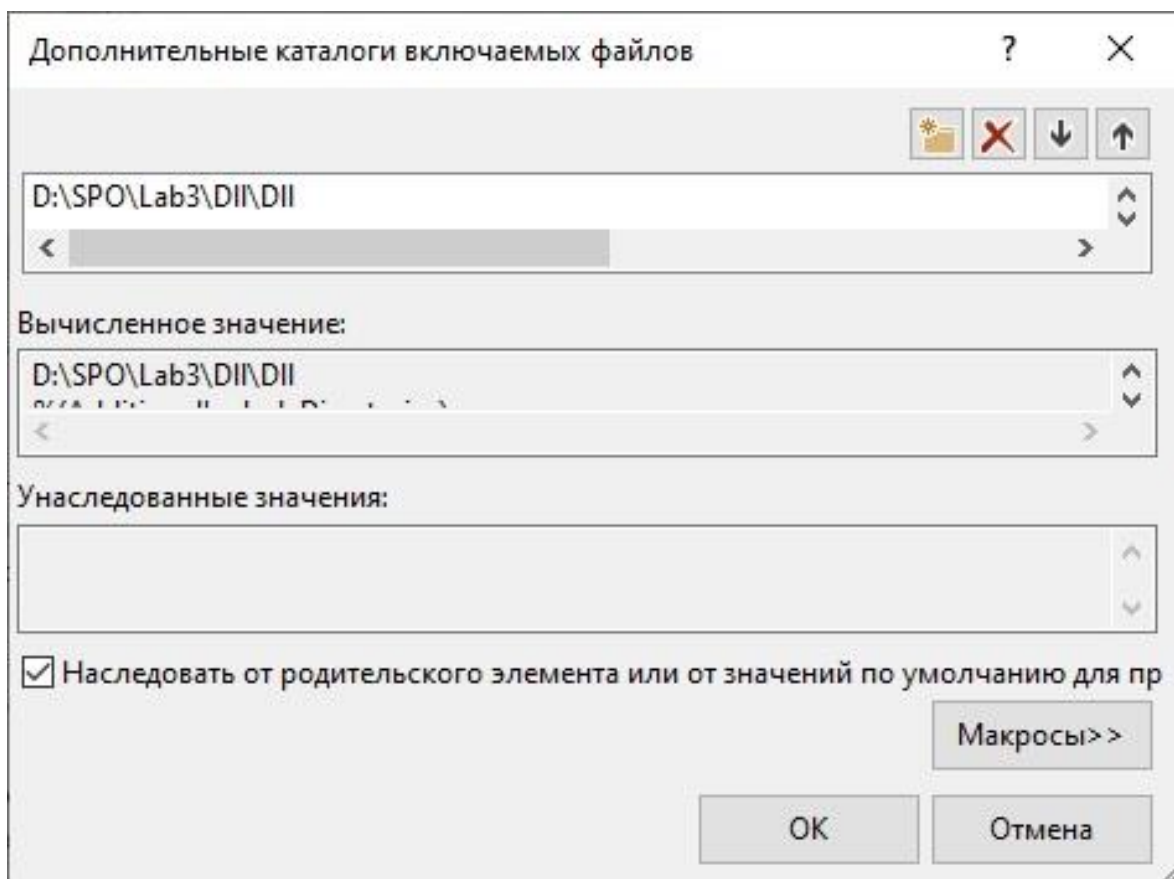
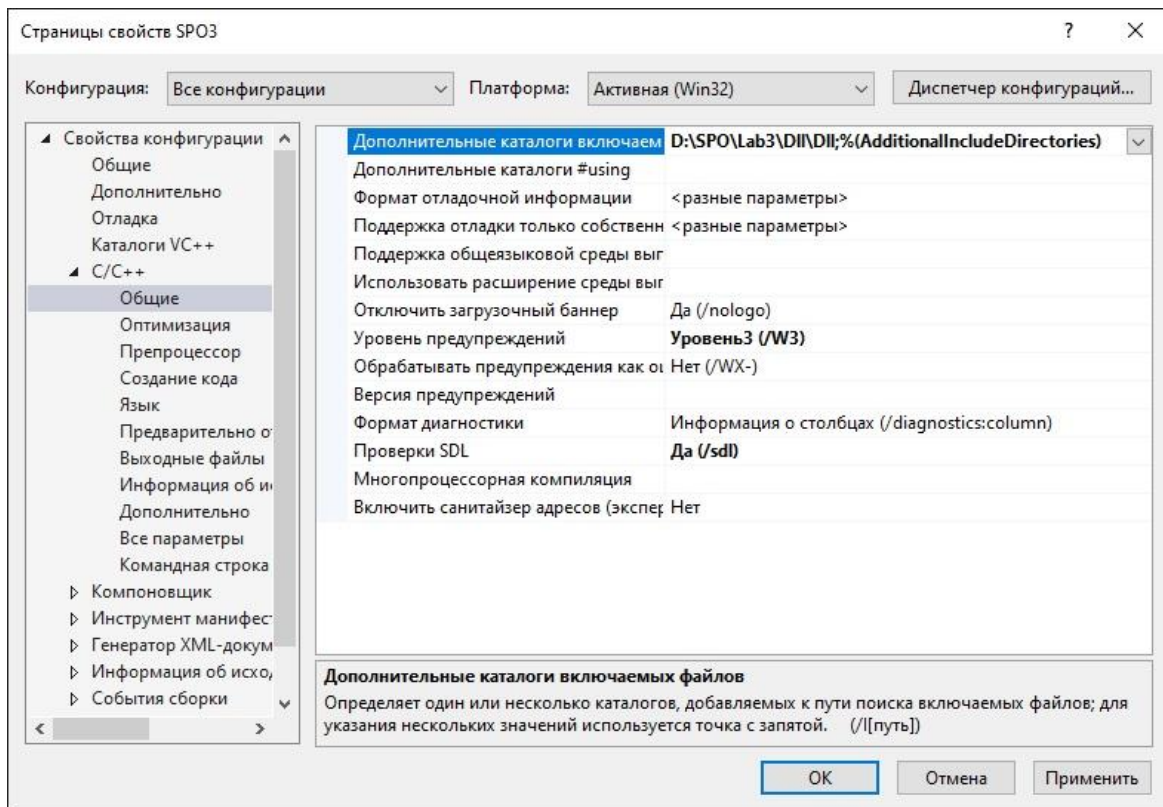
Расположение
D:\SPO\Lab3

Имя решения 1
SPO3

☐ Поместить решение и проект в одном каталоге

Назад Создать

3. Добавление заголовка библиотеки DLL в путь включения



4. Добавление библиотеки импорта DLL в проект

Дополнительные зависимости

dll.lib

Вычисленное значение:

dll.lib
%(AdditionalDependencies)

Унаследованные значения:

kernel32.lib
user32.lib
gdi32.lib

☒ Наследовать от родительского элемента или от значений по умолчанию для пр

Макросы>>

OK Отмена

Страницы свойств SPO3

Конфигурация: Все конфигурации Платформа: Активная (Win32) Диспетчер конфигураций...

Свойства конфигурации

- Общие
- Дополнительно
- Отладка
- Каталоги VC++
- C/C++
- Компоновщик
 - Общие
 - Ввод
 - Файл манифеста
 - Отладка
 - Система
 - Оптимизация
 - Внедренный IDL
 - Метаданные Windows
 - Дополнительно
 - Все параметры
 - Командная строка
- Инструмент манифеста
- Генератор XML-докум
- Информация об исхо
- События сборки

Выходной файл \$(OutDir)\$(TargetName)\$(TargetExt)

Отображать ход выполнения Не задано

Версия

Включить инкрементную компоновку <разные параметры>

Отключить загрузочный баннер Да (/NOLOGO)

Пропустить библиотеку импорта Нет

Зарегистрировать вывод Нет

Перенаправление для пользователей Нет

Дополнительные каталоги библиотек D:\SPO\Lab3\DI\\$(IntDir);%(AdditionalLibraryDirectories)

Компоновать библиотеки из зависим Да

Использовать входные файлы зависи Нет

Состояние ссылки

Запретить привязку Dll

Обрабатывать предупреждения комп

Принудительный вывод файла

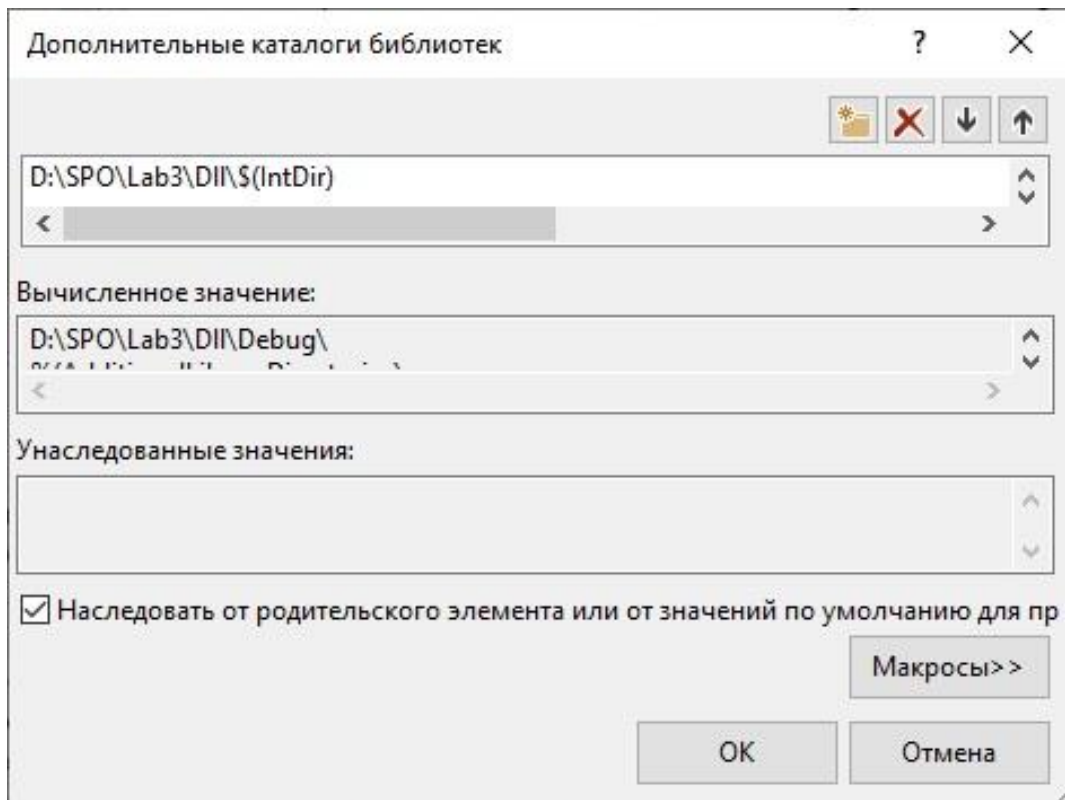
Создать оперативно исправляемый о

Задать атрибуты секции

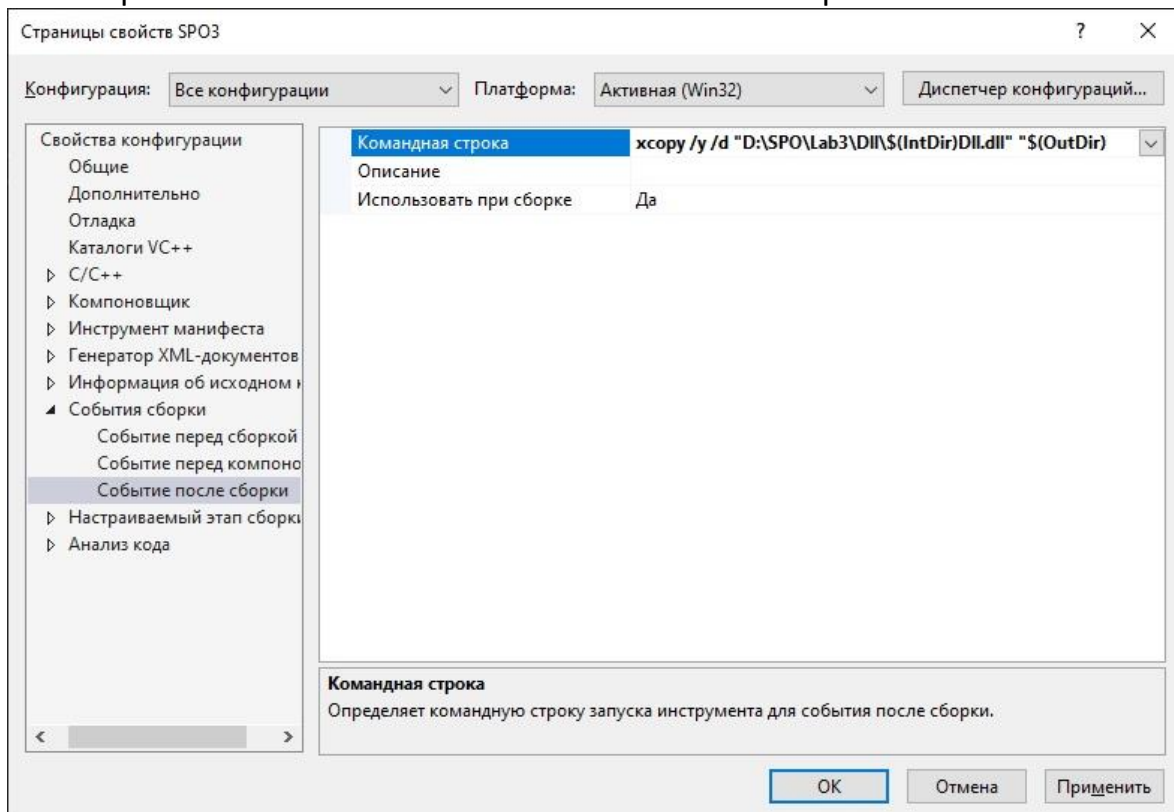
Дополнительные каталоги библиотек

Разрешает пользователю переопределять путь окружения библиотеки. (/LIBPATH:папка)

OK Отмена Применить



5. Копирование библиотеки DLL в событие после сборки



Результат работы программы

```
String: abc
n: 3
Result: abcabcabc
```

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>.

8) Функция, проверяющая существование указанного файла

Lib.cpp

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

```
int exist_file(char* path){
    FILE *a;
    a=fopen(path,"r");
    if(a==NULL)
        return 1;
    else{
        return 0;
        fclose(a);
    }
}
```

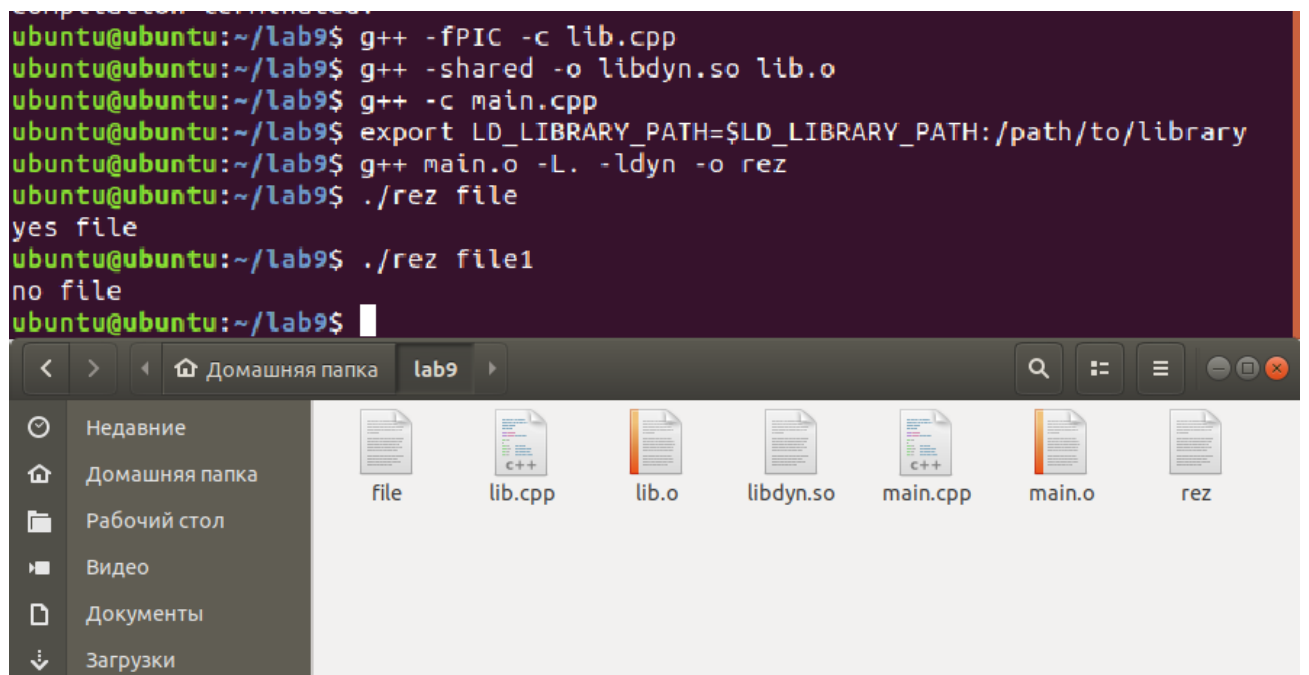
Main.cpp

```
#include <stdio.h>
#include <string>
#include <iostream>
```

```
extern int exist_file(char*);
```

```
int main(int argc, char *argv[]){
    char* path = argv[1];
```

```
    int count = exist_file(path);
    if(count==1)
        std::cout<<"no file"<<std::endl;
    else
        std::cout<<"yes file"<<std::endl;
    return 0;
}
```



Вывод: Закрепил навыки решения задач на строки и манипуляции с файловыми системами, научился проектировать и разрабатывать динамические библиотеки в ОС Windows и Linux

