

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

ОТЧЕТ
По лабораторной работе №3

Выполнил:
Студент 3 курса
группы АС-50
Лещёв Н.И.
Проверила:
Давидюк Ю.И.

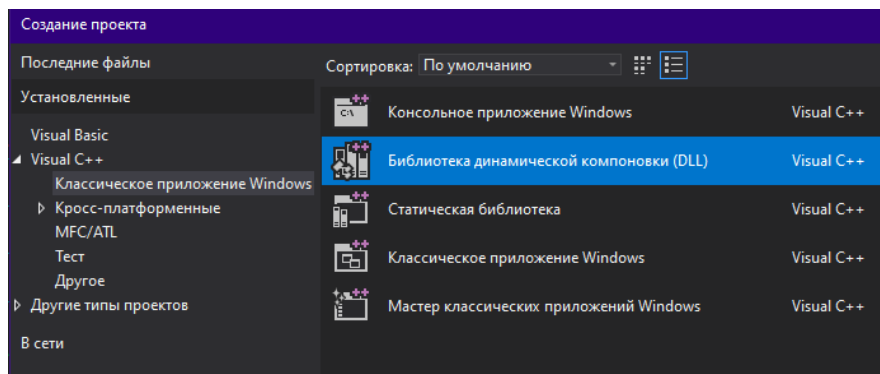
Брест 2020

Вариант – 3 (13)

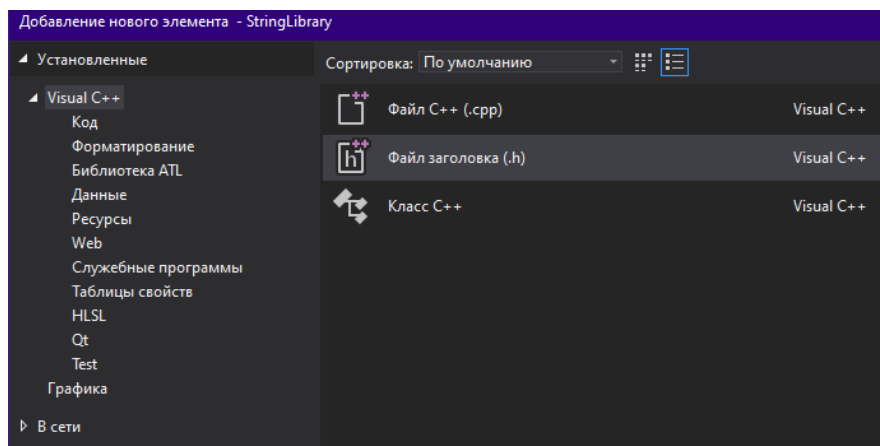
Цель работы: закрепить навыки решения задач на строки и манипуляции с файловыми системами, научиться проектировать и разрабатывать динамические библиотеки в ОС Windows и Linux

Задание 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 «hgfjfs» и точек скрещивания 2 и 3 (для первой и второй строки соответственно) результат скрещивания будет «abfs» и «hgjcd», после итоговой конкатенации – «abfshgjc». Исходный код в:

1. Создаем проект «Библиотека динамической компоновки (DLL)»



2. Создаем заголовочный файл «**StringLibrary.h**», со следующим содержанием



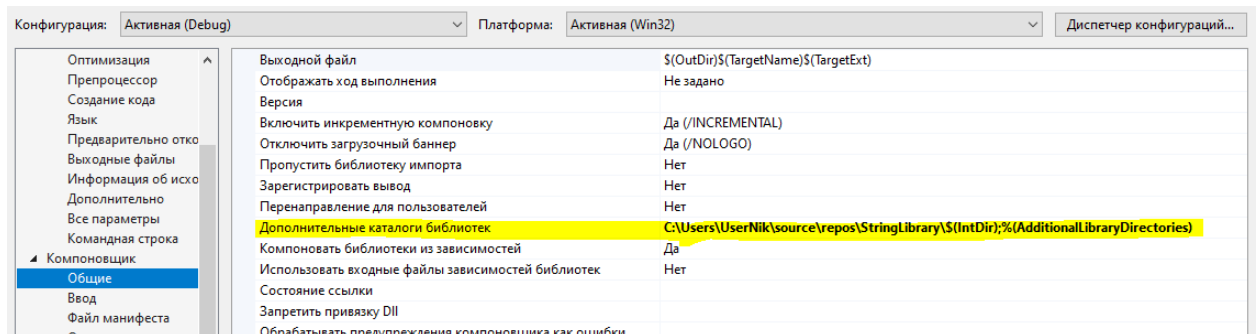
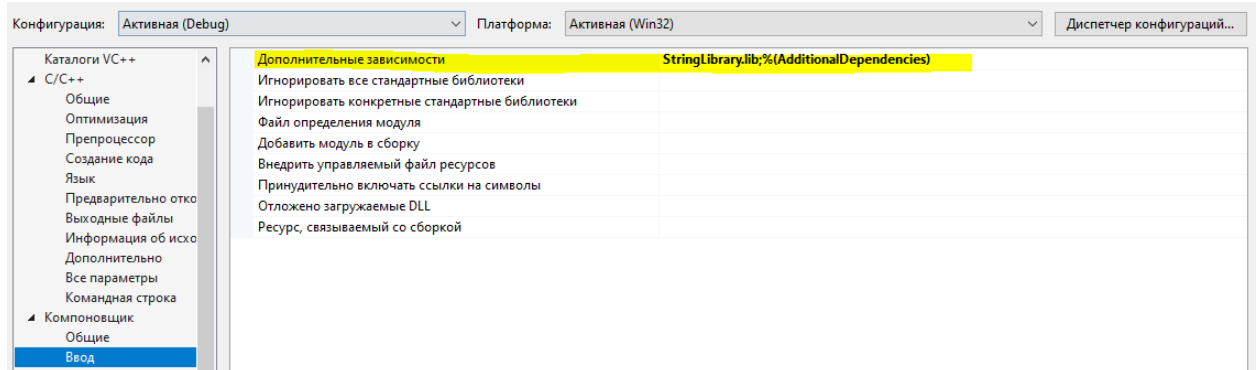
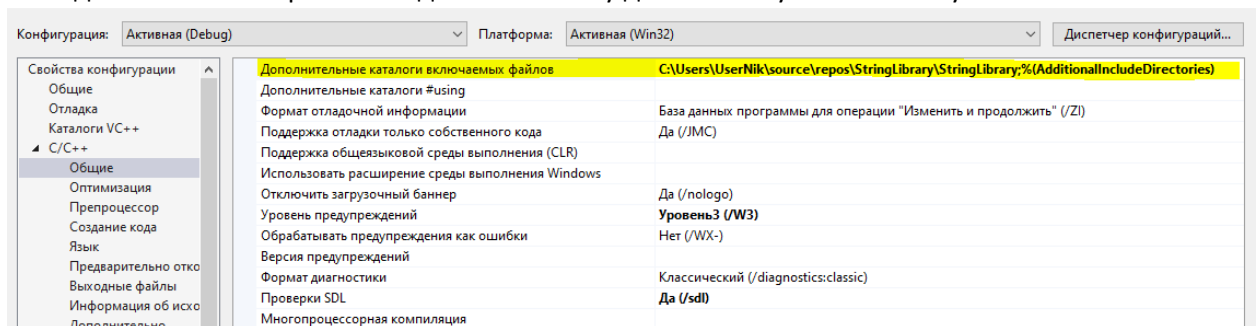
```
#pragma once
#ifdef STRINGLIBRARY_EXPORTS
#define STRINGLIBRARY_API __declspec(dllexport)
#else
#define STRINGLIBRARY_API __declspec(dllimport)
#endif
extern "C" STRINGLIBRARY_API char* crossover(
    char* str1, int pos1, char* str2, int pos2);
```

3. В исходный файл проекта подключаем заголовочный, и описываем функцию, в итоге содержимое файла **StringLibrary.cpp**:

```
#include "stdafx.h"
```

```
#include "StringLibrary.h"
char* crossover(char* str1, int pos1, char* str2, int pos2)
{
    int new_size = strlen(str1) + strlen(str2);
    char* str_cross = new char[new_size + 1];
    int pos_cross = 0;
    int i = 0; //current position on str1
    int j = 0; //current position on str2
    str_cross[new_size] = 0;
    for (; i < pos1; i++, pos_cross++)
        str_cross[pos_cross] = str1[i];
    for (char* handler = str2 + pos2; *handler; handler++, pos_cross++)
        str_cross[pos_cross] = *handler;
    for (; j < pos2; j++, pos_cross++)
        str_cross[pos_cross] = str2[j];
    for (str1 += i; *str1; str1++, pos_cross++)
        str_cross[pos_cross] = *str1;
    return str_cross;
}
```

4. Создаем основной проект и подключаем нашу динамическую библиотеку



5. В итоге наш основной проект будет выглядеть:

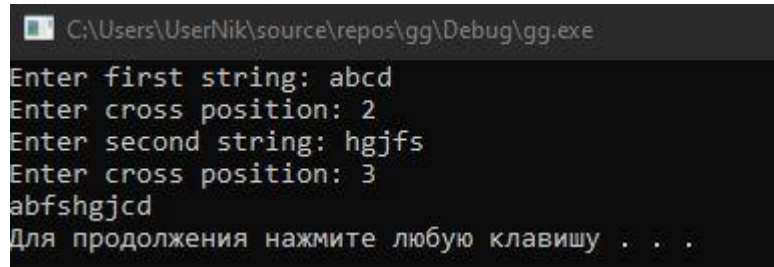
```
#include <iostream>
#include "StringLibrary.h"
int main()
{
    char* msg = new char[1024];
    char* str1 = new char[1024];
```

```

int pos1;
char* str2 = new char[1024];
int pos2;
std::cout << "Enter first string: "; std::cin >> str1;
std::cout << "Enter cross position: "; std::cin >> pos1;
std::cout << "Enter second string: "; std::cin >> str2;
std::cout << "Enter cross position: "; std::cin >> pos2;
msg = crossover(str1, pos1, str2, pos2);
std::cout << msg << std::endl;
//
system("pause");
}

```

И результат выполнения :



```

C:\Users\UserNik\source\repos\gg\Debug\gg.exe
Enter first string: abcd
Enter cross position: 2
Enter second string: hgjfs
Enter cross position: 3
abfshgjcd
Для продолжения нажмите любую клавишу . . .

```

Задание 2: Функция, читающая содержимое указанной директории и выводящая его на экран

Lab3.c

```

#include <dirent.h>
#include <stdlib.h>
#include <stdio.h>
void show_dir(const char* dir)
{
    struct dirent **namelist;
    int n;
    n = scandir(dir, &namelist, 0, alphasort);
    if (n < 0)
        perror("scandir");
    else {
        while(n--) { printf("%s\n", namelist[n]->d_name);
                      free(namelist[n]);
                    }
        free(namelist);
    }
}

```

main.c

```

#include <dirent.h>
#include <stdlib.h>
#include <stdio.h>
void main()
{
    char *dir;
    dir = (char*)malloc(256 * sizeof(char));
    scanf("%s", dir);
    show_dir(dir);
}

```

1.Компилируем наш файл будущей библиотеки

```
nix@nix-VirtualBox:~/lab3go$ gcc -fPIC -c lab3.c
```

2.Создаем динамическую библиотеку

```
nix@nix-VirtualBox:~/lab3go$ gcc -shared -o liblab3.so lab3.o
```

3.Собираем наш основной файл программы

```
nix@nix-VirtualBox:~/lab3go$ gcc -c main.c
```

```
nix@nix-VirtualBox:~/lab3go$ gcc main.o -L. -llab3 -o rezyltdyn
```

4.Смотрим какие библиотеки использует наша программа

```
nix@nix-VirtualBox:~/lab3go$ ldd rezyltdyn
```

```
linux-vdso.so.1 => (0x00007fff1f73b000)
```

```
liblab3.so => not found
```

```
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f8831c7c000)
```

```
/lib64/ld-linux-x86-64.so.2 (0x00007f8832046000)
```

5.Создаем и заполняем файл, откуда динамический линковщик будет брать информацию о директории где искать библиотеку.

```
nix@nix-VirtualBox:/etc/ld.so.conf.d$ sudo touch lab3.conf
```

```
nix@nix-VirtualBox:/etc/ld.so.conf.d$ sudo chmod 777 lab3.conf
```

```
nix@nix-VirtualBox:/etc/ld.so.conf.d$ echo "/home/nix/libs">lab3.conf
```

6.Обновляем изменения конфигурационного файла

```
nix@nix-VirtualBox:~/lab3go$ sudo ldconfig
```

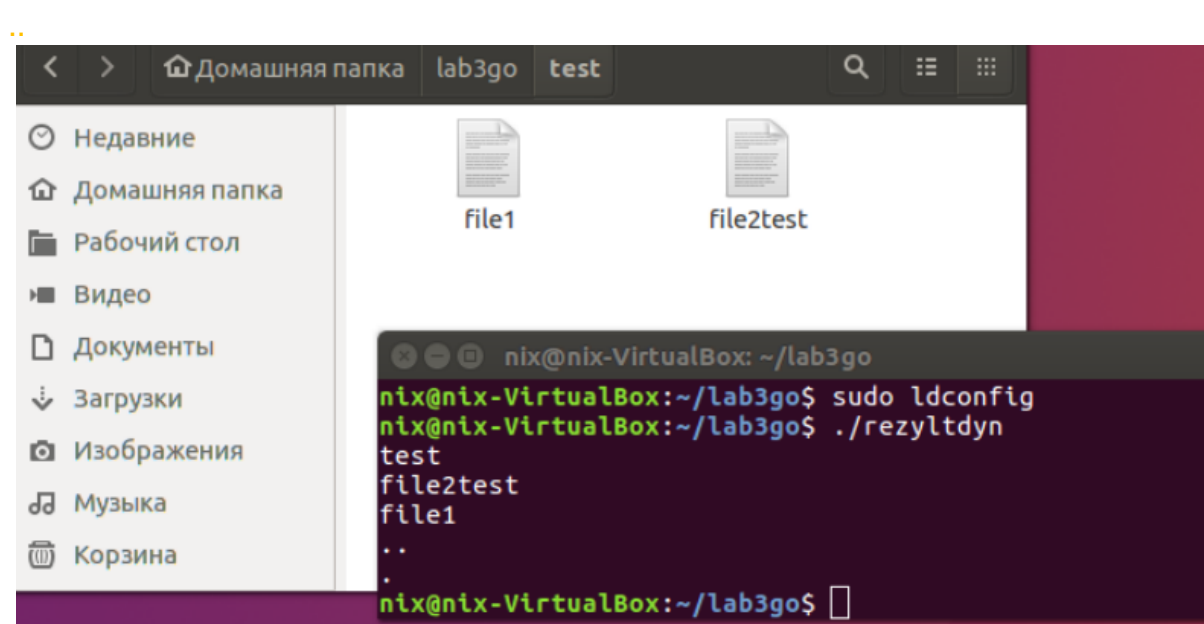
7.Результат запуска программы

```
nix@nix-VirtualBox:~/lab3go$ ./rezyltdyn
```

```
test
```

```
file2test
```

```
file1
```



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