

Министерство образования Республики Беларусь
Учреждение образования
Брестский Государственный Технический университет
Кафедра ИИТ

Лабораторная работа №3
Тема: «Динамические библиотеки в ОС Windows и Linux»

Выполнил:

студ. гр.АС-51

Иванюк С.С.

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.

Ход работы

Вариант 18

Задание - Windows

8) 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.

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

Заголовочный файл в библиотеке DLL:

```
#pragma once
#ifdef CONCATENATION_EXPORTS
#define CONCATENATION_API __declspec(dllexport)
#else
#define CONCATENATION_API __declspec(dllimport)
#endif
extern "C" CONCATENATION_API void concatenation(char str[], int n);
```

Функция в библиотеке DLL:

```
#include "pch.h"
#include <iostream>
#include <string.h>
#include "concatenation.h"
#pragma warning(disable : 4996)

char *concatenation(char str[], int n) {
    char s[1000];
    strcpy_s(s, str);
    for (int i = 0; i < n; i++) {
        strcat(str, s);
    }
    return str;
}
```

Код приложения, использующий библиотеку DLL:

```
#include "pch.h"
#include <iostream>
#include "concatenation.h"
int main() {
    int n;
    char str[1000];
    std::cout << "Enter str: ";
    std::cin.getline(str, 100);
    std::cout << "Enter n: ";
    std::cin >> n;
    std::cout << "The result of concatenation: ";
    std::cout << concatenation(str, n);
    return 0;
}
```

Вывод:

```
Enter str: It's my lab3!  
Enter n: 3  
The result of concatenation: It's my lab3! It's my lab3! It's my lab3!
```

Подключение библиотеки:

1.) Добавление заголовка библиотеки DLL в путь включения:

Страницы свойств concatenationClient ? X

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

Свойства конфигурации
Общие
Дополнительно
Отладка
Каталоги VC++
C/C++
Общие
Оптимизация
Препроцессор
Создание кода
Язык
Предварительно отк
Выходные файлы
Информация об исхо
Дополнительно
Все параметры
Командная строка
Компоновщик
Инструмент манифеста
Генератор XML-докумен
Информация об исходн
События сборки
Настраиваемый этап сб

Дополнительные каталоги включает C:\Users\Ceprei\source\repos\concatenation\concatenatio
Дополнительные каталоги #using
Формат отладочной информации <разные параметры>
Поддержка отладки только собствен <разные параметры>
Поддержка общезыковой среды ви
Использовать расширение среды ви
Отключить загрузочный баннер Да (/nologo)
Уровень предупреждений **Уровень3 (/W3)**
Обрабатывать предупреждения как Нет (/WX-)
Версия предупреждений
Формат диагностики Информация о столбцах (/diagnostics:column)
Проверки SDL Да (/sdl)
Многопроцессорная компиляция
Включить санитайзер адресов (эксп Нет

Дополнительные каталоги включаемых файлов
Определяет один или несколько каталогов, добавляемых к пути поиска включаемых файлов; для указания нескольких значений используется точка с запятой. (/I[путь])

OK Отмена Применить

2.) Добавление библиотеки импорта DLL в проект:

Страницы свойств concatenationClient ? X

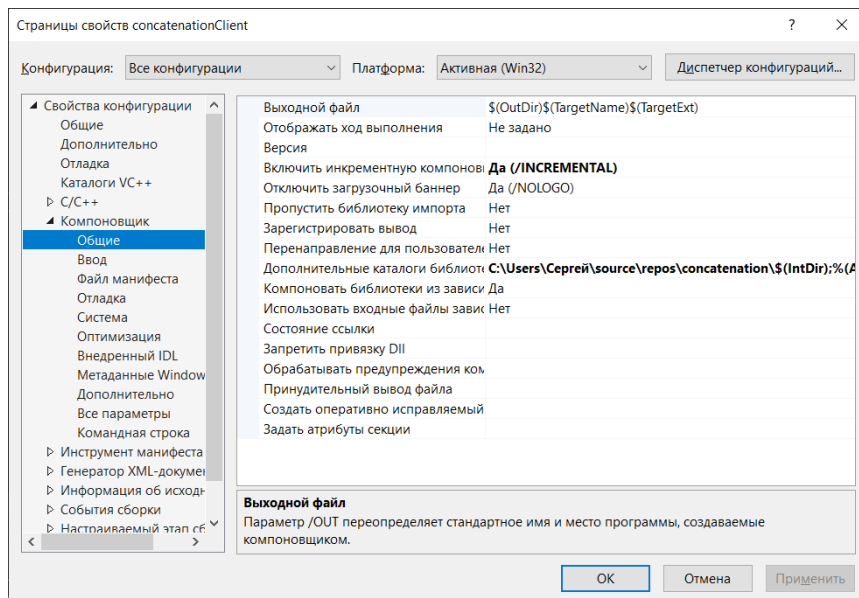
Конфигурация: Все конфигурации Платформа: Активная (Win32) Диспетчер конфигураций...

Свойства конфигурации
Общие
Дополнительно
Отладка
Каталоги VC++
C/C++
Компоновщик
Общие
Ввод
Файл манифеста
Отладка
Система
Оптимизация
Внедренный IDL
Метаданные Window
Дополнительно
Все параметры
Командная строка
Инструмент манифеста
Генератор XML-докумен
Информация об исходн
События сборки
Настраиваемый этап сб

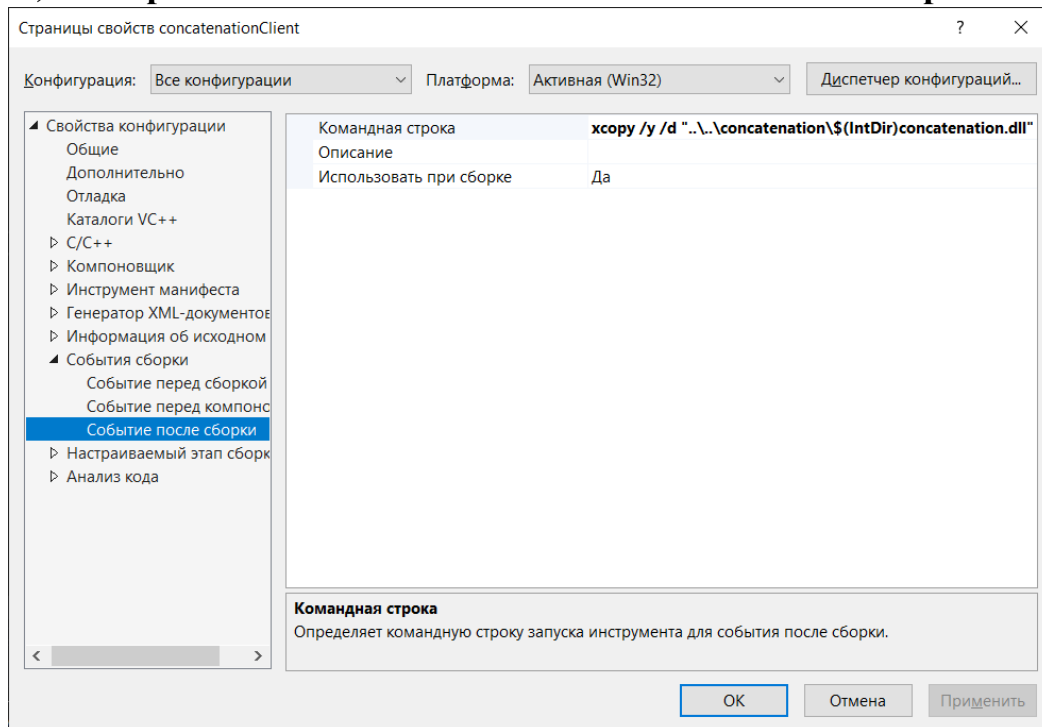
Дополнительные зависимости **concatenation.lib;%(AdditionalDependencies)**
Игнорировать все стандартные биб
Игнорировать конкретные стандарт
Файл определения модуля
Добавить модуль в сборку
Внедрить управляемый файл ресурс
Принудительно включать ссылки на
Отложено загружаемые DLL
Ресурс, связываемый со сборкой

Дополнительные зависимости
Задаёт дополнительные элементы, которые нужно добавить в командную строку компоновки.
(t.e. kernel32.lib)

OK Отмена Применить



3.) Копирование библиотеки DLL в событие после сборки:



Задание - Linux

8) Разработать динамическую библиотеку `so`, включающую функцию (функции), реализующую следующий функционал. Продемонстрировать ее подключение и использование. Функция, проверяющая существование указанного файла.

Kod check.c:

```
#include <stdio.h>
#include <unistd.h>

void check(const char *fname) {
    if ( !access (fname, 0) )
        printf("File present\n");
    else
        printf("File not Found\n");
}
```

Kod lab3.c:

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

```
extern void check();
int main() {
    char fname[100];
    printf("Enter the file name: ");
    scanf("%s", fname);
    check(fname);
    return 0;
}
```

Вывод:

```
sergey@sergey:~$ gcc -fPIC -c check.c
sergey@sergey:~$ gcc -shared -o libcheck.so check.o
sergey@sergey:~$ gcc -c lab3.c
sergey@sergey:~$ gcc lab3.o -L. -lcheck -o rezult
sergey@sergey:~$ ./rezult
Enter the file name: lab3.c
File present
```

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