

Московский Авиационный Институт  
(Национальный Исследовательский Университет)  
Факультет информационных технологий и прикладной математики  
Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу**  
**«Операционные системы»**

Студент: Мусаелян Ярослав  
Группа: М8О-207Б-21  
Вариант: 19  
Преподаватель: Миронов Евгений Сергеевич  
Оценка: \_\_\_\_\_  
Дата: \_\_\_\_\_  
Подпись: \_\_\_\_\_

Москва, 2022

## **Содержание**

- 1 Репозиторий
- 2 Постановка задачи
- 3 Общие сведения о программе
- 4 Общий метод и алгоритм решения
- 5 Исходный код
- 6 Демонстрация работы программы
- 7 Выводы

## Репозиторий

<https://github.com/YMusaelyan/os>

### Постановка задачи

#### Цель работы

Приобретение практических навыков в создании динамических библиотек, в создании программ, которые используют функции динамических библиотек, в работе со сборочной системой

#### Задание

Требуется создать динамические библиотеки, которые реализуют определенный функционал. Далее использовать данные библиотеки 2-мя способами:

- 1 Во время компиляции (на этапе «линковки»/linking)
- 2 Во время исполнения программы. Библиотеки загружаются в память с помощью интерфейса ОС для работы с динамическими библиотеками

В конечном итоге, в лабораторной работе необходимо получить следующие части:

- Динамические библиотеки, реализующие контракты, которые заданы вариантом;
- Тестовая программа (*программа №1*), которая использует одну из библиотек, используя знания полученные на этапе компиляции;
- Тестовая программа (*программа №2*), которая загружает библиотеки, используя только их местоположение и контракты.

Провести анализ двух типов использования библиотек.

Пользовательский ввод для обеих программ должен быть организован следующим образом:

- 1 Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для *программы №2*). Можно реализовать лабораторную работу без данной функции, но максимальная оценка в этом случае будет «хорошо»;
  - 2 «1 arg1 arg2 ... argN», где после «1» идут аргументы для первой функции, предусмотренной контрактами. После ввода команды происходит вызов первой функции, и на экране появляется результат её выполнения;
- «2 arg1 arg2 ... argM», где после «2» идут аргументы для второй функции, предусмотренной контрактами. После ввода команды происходит вызов второй функции, и на экране появляется результат её выполнения.

19 вариант

3	Подсчёт количества простых чисел на отрезке [A, B] (A, B - натуральные)	Int <u>PrimeCount</u> (int A, int B)	Наивный алгоритм. Проверить делимость текущего числа на все предыдущие числа.	<u>Решето Эратосфена</u>
---	-------------------------------------------------------------------------	-----------------------------------------	-------------------------------------------------------------------------------	--------------------------

7	Подсчет площади плоской геометрической фигуры по двум сторонам	Float Square(float A, float B)	Фигура прямоугольни к	Фигура прямоугольн ый треугольник
3	CMake	Возможность сборки всех таргетов с ASAN без переопределения CMake флагов, если указана соответствующая переменная и ОС имеет поддержку ASAN.		

## Общие сведения о программе

Программа компилируется из файлов main.cpp, child.cpp. В программе используются следующие системные вызовы:

- 1 dlopen()- загружает динамическую библиотеку, имя которой указано в строке *filename*, и возвращает прямой указатель на начало динамической библиотеки.
- 2 dlsym()- использует указатель на динамическую библиотеку, возвращаемую dlopen, и оканчивающееся нулем символьное имя, а затем возвращает адрес, указывающий, откуда загружается этот символ
- 3 dlclose() - уменьшает на единицу счетчик ссылок на указатель динамической библиотеки *handle*. Если нет других загруженных библиотек, использующих ее символы и если счетчик ссылок принимает нулевое значение, то динамическая библиотека выгружается.

## Общий метод и алгоритм решения

Реализуем библиотеки lib1.cpp и lib2.cpp, где в первой будет функция вычисления простого числа простым перебором и функция вычисления площади, а во втором функция вычисление простого числа будет с помощью Решета Эратосфена, а функция площади будет считать площадь прямоугольного треугольника, а так же объявим данные функции внутри файла lib.h.

Для первого случая получаем lib.h, и пользуемся реализованными функциями, как обычная библиотека, но в отличии от нее происходит на этапе сборки программы.

Для второго случая мы используем системные вызовы из библиотеки <dlfcn.h> и открываем динамические библиотеки по названию и загружаем из них необходимые функции. Так же реализуем небольшой пользовательский ввод с помощью цикла и условий.

## Исходный код

**lib.h**

```
#ifndef __LIB_H__
```

```
#define __LIB_H__
```

```
extern "C" int PrimeCount(int A, int B);
```

```
extern "C" float Square(float A, float B);
```

```
#endif
```

```
lib1.cpp
```

```
#include<iostream>
```

```
using namespace std;
```

```
extern "C" int PrimeCount(int A, int B) {  
    int cnt = 0, flag = 1;  
    for(int i = A; i <= B; i++) {  
        flag = 1;  
        if (i <= 1) {  
            continue;  
        }  
        for(int j = 2; j*j <= i; j++) {  
            if(i % j == 0) {  
                flag = 0;  
            }  
        }  
        if (flag == 1) {  
            cnt++;  
        }  
    }  
    return cnt;  
}
```

```
extern "C" float Square(float A, float B) {  
    return A * B;  
}
```

```
lib2.cpp
```

```
#include<iostream>
```

```
using namespace std;
```

```
extern "C" int PrimeCount(int A, int B) {  
    int mas[B], cnt = 0;  
    for(int i = A; i <= B; i++) {
```

```

        if(B <= 1) {
            return cnt;
        }
        if(i <= 1) {
            continue;
        }
        mas[i] = i;
    }
    mas[1] = 0;
    for(int i = 2; i <= B; i++) {
        if(mas[i] != 0) {
            for(int j = i * 2; j <= B; j += i){
                mas[j] = 0;
            }
        }
    }
    for(int i = A; i <= B; i++) {
        if(i <= 1) {
            continue;
        }
        if (mas[i] != 0) {
            cnt++;
        }
    }
    return cnt;
}

```

```

extern "C" float Square(float A, float B) {
    return (A * B) / 2.f;
}

```

### **main1.cpp**

```

#include<iostream>
#include"../include/lib.h"

```

```

using namespace std;

```

```

int main()
{

```

```

int k;
cout << "Enter 1 or 2 to select a function or -1 to exit" << endl;

while(cin >> k) {

    if (k == 1) {
        int A, B;
        cout << "Enter section [A,B]" << endl;
        cin >> A >> B;
        cout<<"Count of prime numbers:" <<
PrimeCount(A, B) << endl;
    }

    if (k == 2) {
        float A, B;
        cout << "Enter side lengths A and B" << endl;
        cin >> A >> B;
        cout << "Square:" << Square(A, B) << endl;
    }

    if (k == -1) {
        cout << "Exit" << endl;
        return 0;
    }

    cout << "Enter 1 or 2 to select a function or -1 to exit" <<
endl;

}

}

main2.cpp
#include<iostream>
#include<dlfcn.h>
#include"lib.h"
#include<string>

using namespace std;

```

```

int main()
{
    string lib1 = "./liblib1.so";
    string lib2 = "./liblib2.so";

    void *curlib = dlopen(lib1.c_str(), RTLD_LAZY);

    int (*PrimeCount)(int A, int B);
    float (*Square)(float A, float B);

    PrimeCount = (int (*)(int, int))dlsym(curlib, "PrimeCount");
    Square = (float (*)(float, float))dlsym(curlib, "Square");

    int k;
    int number_lib = 1;
    cout << "Enter 1 or 2 to select a function or -1 to exit or 0 switch
to library" << endl;

    while(cin >> k) {

        if (k == 0) {
            dlclose(curlib);
            if (number_lib == 1) {
                cout << "You switched to library number
2" << endl;

                curlib = dlopen(lib2.c_str(),
RTLD_LAZY);

                number_lib = 2;

            } else {
                cout << "You switched to library number
1" << endl;

                curlib = dlopen(lib1.c_str(),
RTLD_LAZY);

                number_lib = 1;
            }
            PrimeCount = (int (*)(int, int))dlsym(curlib,
"PrimeCount");

```



```

        Square = (float (*)(float, float))dlsym(curlib, "Square");
    }

    if (k == 1) {
        int A, B, Ans1;
        cout << "Enter section [A,B]" << endl;
        cin >> A >> B;
        Ans1 = PrimeCount(A, B);
        cout<<"Count of prime numbers:" << Ans1 <<
endl;

    }

    if (k == 2) {
        float A, B, Ans2;
        cout << "Enter side lengths A and B" << endl;
        cin >> A >> B;
        Ans2 = Square(A, B);
        cout << "Square:" << Ans2 << endl;
    }

    if (k == -1) {
        cout << "Exit" << endl;
        return 0;
    }

    cout << "Enter 1 or 2 to select a function or -1 to exit or 0
switch to library" << endl;

}

```

```

}

```

### **CmakeList.txt**

```

cmake_minimum_required(VERSION 3.8 FATAL_ERROR)
project(main LANGUAGES CXX)

set(BUILD_WITH_ASAN 1)
set(CMAKE_CXX_FLAGS "-fsanitize=address -ldl -lrt -lpthread")

add_library(
    lib1 SHARED
    ./include/lib.h

```

```

        ./src/lib1.cpp
    )
    add_library(
        lib2 SHARED
        ./include/lib.h
        ./src/lib2.cpp
    )

    add_executable(main1 ./src/main1.cpp)
    target_include_directories(main1 PRIVATE ./include)
    target_link_libraries(main1 PRIVATE lib1 m)

    add_executable(main2 ./src/main1.cpp)
    target_include_directories(main2 PRIVATE ./include)
    target_link_libraries(main2 PRIVATE lib2 m)

    add_executable(main ./src/main2.cpp)
    target_include_directories(main PRIVATE ./include m)

    if (${BUILD_WITH_ASAN})
        message("-- Adding sanitizers")
        target_compile_options(main PRIVATE)
        target_link_options(main PRIVATE)
        target_compile_options(main1 PRIVATE)
        target_link_options(main1 PRIVATE)
        target_compile_options(main2 PRIVATE)
        target_link_options(main2 PRIVATE)
    endif()

```

## Демонстрация работы программы

```

yarik@asus:~/os/os/lab5$ ./main1
Enter 1 or 2 to select a function or -1 to exit
1
Enter section [A,B]
1 10
Count of prime numbers:4
Enter 1 or 2 to select a function or -1 to exit
2
Enter side lengths A and B
1.2
1.2
Square:1.44
Enter 1 or 2 to select a function or -1 to exit
-1
Exit

```

```
yarik@asus:~/os/os/lab5$ ./main
Enter 1 or 2 to select a function or -1 to exit or 0 switch to library
1
Enter section [A,B]
1 10
Count of prime numbers:4
Enter 1 or 2 to select a function or -1 to exit or 0 switch to library
2
Enter side lengths A and B
1.2 1.2
Square:1.44
Enter 1 or 2 to select a function or -1 to exit or 0 switch to library
0
You switched to library number 2
Enter 1 or 2 to select a function or -1 to exit or 0 switch to library
2
Enter side lengths A and B
1.2 1.2
Square:0.72
Enter 1 or 2 to select a function or -1 to exit or 0 switch to library
-1
Exit
```

## Выводы

В результате проделанной работы я приобрел практические навыки в создании динамических библиотек, в создании программ, которые используют функции динамических библиотек, в работе со сборочной системой