

Московский авиационный институт
(Национальный исследовательский университет)
Факультет "Информационные технологии и прикладная математика"
Кафедра "Вычислительная математика и программирование"

**Лабораторная работа №4 по курсу
“Операционные системы”**

Студент: Былькова Кристина Алексеевна

Группа: М8О-208Б-22

Преподаватель: Миронов Евгений Сергеевич

Вариант: 2

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2023

Содержание

1	Репозиторий	3
2	Цель работы	3
3	Задание	3
4	Описание работы программы	3
5	Исходный код	4
6	Тесты	8
7	Консоль	9
8	Запуск тестов	9
9	Выводы	11

1 Репозиторий

https://github.com/kr1st1na0/OS_labs

2 Цель работы

Приобретение практических навыков в:

- Создании динамических библиотек
- Создании программ, которые используют функции динамических библиотек

3 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.

4 Описание работы программы

Функции, написанные в результате выполнения лабораторной работы:

- Расчет производной функции $\cos(x)$ в точке A с приращением δx двумя реализациями
- Подсчет количества простых чисел на отрезке $[A, B]$ (A, B - натуральные) наивным алгоритмом и решето Эратосфена

В ходе выполнения лабораторной работы я использовала следующие системные вызовы:

- `dlopen` - открытие динамического объекта
- `dlclose` - закрытие динамического объекта

5 Исходный код

utils.hpp

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <dlfcn.h>
4
5 using SinIntegralFunc = float (*)(float, float, float);
6 using PrimeCountFunc = int (*)(int, int);
7
8 void* LoadLibrary(const char *libraryName);
9 void UnloadLibrary(void* handle);
```

realizations.hpp

```
1 #pragma once
2
3 #include <iostream>
4 #include <cmath>
5
6 #ifdef __cplusplus
7 extern "C" {
8 #endif
9
10 constexpr int NUM_POINTS = 3000; //
11 const float PI = 3.1415926535;
12
13 float SinIntegral(float a, float b, float e);
14 int PrimeCount(int a, int b);
15
16 #ifdef __cplusplus
17 }
18 #endif
```

utils.cpp

```
1 #include "utils.hpp"
2
3 void* LoadLibrary(const char *libraryName) {
4     void* handle = dlopen(libraryName, RTLD_LAZY);
5     if (!handle) {
6         std::cerr << "Couldn't load the library: " << dlerror() <<
7             std::endl;
8         exit(EXIT_FAILURE);
9     }
10    return handle;
11 }
12
13 void UnloadLibrary(void* handle) {
14     if (dlclose(handle) != 0) {
15         std::cerr << "Couldn't unload the library: " << dlerror()
16             << std::endl;
17         exit(EXIT_FAILURE);
18     }
19 }
```

realization1.cpp

```
1 #include "realizations.hpp"
2
3 extern "C" float SinIntegral(float a, float b, float e) {
```

```

4     float integral = 0.0;
5     e = (b - a) / NUM_POINTS;
6     for (int i = 0; i <= NUM_POINTS; ++i) {
7         integral = integral + e * sin(a + e * (i - 0.5));
8     }
9     return integral;
10 }
11
12 extern "C" int PrimeCount(int a, int b) {
13     int count = 0;
14     bool flag = true;
15     for (int i = a; i <= b; ++i) {
16         if (i <= 1) {
17             continue;
18         }
19         for (int j = 2; j < i; ++j) {
20             if (i % j == 0) {
21                 flag = false;
22                 break;
23             }
24         }
25         if (flag) {
26             ++count;
27         }
28         flag = true;
29     }
30     return count;
31 }

```

realization2.cpp

```

1 #include "realizations.hpp"
2 #include <vector>
3
4 extern "C" float SinIntegral(float a, float b, float e) {
5     float integral = 0.0;
6     e = (b - a) / NUM_POINTS;
7     for (int i = 1; i < NUM_POINTS; ++i) {
8         float x1 = a + i * e;
9         float x2 = a + (i + 1) * e;
10        integral += 0.5 * e * (sin(x1) + sin(x2));
11    }
12    return integral;
13 }
14
15 extern "C" int PrimeCount(int a, int b) {
16     int count = 0;
17     std::vector<int> numbers;
18     numbers.reserve(b + 1);
19     for (int i = 0; i <= b; ++i) {
20         numbers.push_back(i);
21     }
22     for (int i = 2; i <= b; ++i) {
23         if (numbers[i] != 0) {
24             if (numbers[i] >= a && numbers[i] <= b) {
25                 ++count;
26             }
27             for (int j = i * i; j <= b; j += i) {
28                 numbers[j] = 0;
29             }

```

```

30     }
31 }
32     return count;
33 }

```

dynamic_main.cpp

```

1 #include "utils.hpp"
2
3 int main() {
4     const char *pathToLib1 = getenv("PATH_TO_LIB1");
5     const char *pathToLib2 = getenv("PATH_TO_LIB2");
6     // bash: export PATH_TO_LIB1="/home/kristinab/ubuntu_main/
7     OS_labs/build/lab4/librealization1.so"
8     // bash: export PATH_TO_LIB2="/home/kristinab/ubuntu_main/
9     OS_labs/build/lab4/librealization2.so"
10
11     void* libraryHandle = LoadLibrary(pathToLib1);
12     SinIntegralFunc SinIntegral = (SinIntegralFunc)dlsym(
13     libraryHandle, "SinIntegral");
14     PrimeCountFunc PrimeCount = (PrimeCountFunc)dlsym(
15     libraryHandle, "PrimeCount");
16
17     std::string command;
18     while(true) {
19         std::cout << "Enter the command (0 - switch realization, e
20         - exit): ";
21         std::cin >> command;
22         if (command == "e") {
23             break;
24         } else if (command == "0") {
25             std::cout << "Enter the library (1 or 2): ";
26             std::cin >> command;
27             if (command == "1") {
28                 libraryHandle = LoadLibrary(pathToLib1);
29             } else if (command == "2") {
30                 libraryHandle = LoadLibrary(pathToLib2);
31             } else {
32                 std::cout << "Invalid library" << std::endl;
33             }
34             SinIntegral = (SinIntegralFunc)dlsym(libraryHandle, "
35             SinIntegral");
36             PrimeCount = (PrimeCountFunc)dlsym(libraryHandle, "
37             PrimeCount");
38         } else {
39             if (command == "1") {
40                 std::cout << "SinIntegral function:" << std::endl;
41                 float arg1, arg2, arg3;
42                 std::cin >> arg1 >> arg2 >> arg3;
43                 float result = SinIntegral(arg1, arg2, arg3);
44                 std::cout << "Result of integral = " << result <<
45                 std::endl;
46             } else if (command == "2") {
47                 std::cout << "PrimeCount function:" << std::endl;
48                 int arg1, arg2;
49                 std::cin >> arg1 >> arg2;
50                 int result = PrimeCount(arg1, arg2);
51                 std::cout << "Count of prime numbers = " << result
52                 << std::endl;
53             } else {

```

```

45         std::cout << "Invalid command" << std::endl;
46     }
47 }
48 }
49 UnloadLibrary(libraryHandle);
50 return 0;
51 }

static _main.cpp

1 #include "realizations.hpp"
2
3 #include <iostream>
4
5 void Task(const std::string& command) {
6     if (command == "1") {
7         float arg1, arg2, arg3;
8         std::cin >> arg1 >> arg2 >> arg3;
9         float result = SinIntegral(arg1, arg2, arg3);
10        std::cout << "Result of integral = " << result << std::
endl;
11    } else if (command == "2") {
12        int arg1, arg2;
13        std::cin >> arg1 >> arg2;
14        int result = PrimeCount(arg1, arg2);
15        std::cout << "Count of prime numbers = " << result << std
::endl;
16    } else {
17        std::cout << "Invalid command" << std::endl;
18    }
19 }
20
21 int main() {
22     std::string command;
23     while(true) {
24         std::cout << "Enter the command (0 - exit): ";
25         std::cin >> command;
26         if (command == "0") {
27             break;
28         }
29         Task(command);
30     }
31     return 0;
32 }

```

6 Тесты

```
1 #include "gtest/gtest.h"
2 #include "realizations.hpp"
3
4 TEST(fourthLabTest, SinIntegralStaticTest1) {
5     float result = SinIntegral(0, PI, 0.01);
6     EXPECT_FLOAT_EQ(result, 2);
7 }
8
9 TEST(fourthLabTest, SinIntegralStaticTest2) {
10    float result = SinIntegral(0, PI/2, 0.01);
11    EXPECT_FLOAT_EQ(result, 1);
12 }
13
14 TEST(fourthLabTest, SinIntegralStaticTest3) {
15    float result = SinIntegral(0, PI/3, 0.01);
16    EXPECT_FLOAT_EQ(result, 0.49999982);
17 }
18
19 TEST(fourthLabTest, PrimeCountStaticTest) {
20    float result = PrimeCount(3, 15);
21    EXPECT_FLOAT_EQ(result, 5);
22 }
23
24 int main(int argc, char **argv) {
25     testing::InitGoogleTest(&argc, argv);
26     return RUN_ALL_TESTS();
27 }

1 #include "gtest/gtest.h"
2 #include "realizations.hpp"
3
4 TEST(fourthLabTest, SinIntegralStaticTest1) {
5     float result = SinIntegral(0, PI, 0.01);
6     EXPECT_FLOAT_EQ(result, 2);
7 }
8
9 TEST(fourthLabTest, SinIntegralStaticTest2) {
10    float result = SinIntegral(0, PI/2, 0.01);
11    EXPECT_FLOAT_EQ(result, 0.9999997);
12 }
13
14 TEST(fourthLabTest, SinIntegralStaticTest3) {
15    float result = SinIntegral(0, PI/3, 0.01);
16    EXPECT_FLOAT_EQ(result, 0.49999979);
17 }
18
19 TEST(fourthLabTest, PrimeCountStaticTest) {
20    float result = PrimeCount(0, 10);
21    EXPECT_FLOAT_EQ(result, 4);
22 }
23
24 int main(int argc, char **argv) {
25     testing::InitGoogleTest(&argc, argv);
26     return RUN_ALL_TESTS();
27 }
```


7 Консоль

```
kristinab@LAPTOP-SFU9B1F4:~/ubuntu_main/OS_labs/build/lab4$ ./static_main
Enter the command (0 - exit): 1
0 3.14 0.01
Result of integral = 2
Enter the command (0 - exit): 2
3 15
Count of prime numbers = 5
Enter the command (0 - exit): 0
kristinab@LAPTOP-SFU9B1F4:~/ubuntu_main/OS_labs/build/lab4$ ./dynamic_main
Enter the command (0 - switch realization, e - exit): 0
Enter the library (1 or 2): 1
Enter the command (0 - switch realization, e - exit): 1
SinIntegral function:
0 3.14 0.01
Result of integral = 2
Enter the command (0 - switch realization, e - exit): 2
PrimeCount function:
0 10
Count of prime numbers = 4
Enter the command (0 - switch realization, e - exit): 0
Enter the library (1 or 2): 2
Enter the command (0 - switch realization, e - exit): 1
SinIntegral function:
0 3.14 0.01
Result of integral = 2
Enter the command (0 - switch realization, e - exit): 2
PrimeCount function:
0 10
Count of prime numbers = 4
Enter the command (0 - switch realization, e - exit): e
```

8 Запуск тестов

```
kristinab@LAPTOP-SFU9B1F4:~/ubuntu_main/OS_labs/build/tests$ ./lab4_test1
[=====] Running 4 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 4 tests from fourthLabTest
[ RUN      ] fourthLabTest.SinIntegralStaticTest1
[      OK  ] fourthLabTest.SinIntegralStaticTest1 (0 ms)
[ RUN      ] fourthLabTest.SinIntegralStaticTest2
[      OK  ] fourthLabTest.SinIntegralStaticTest2 (0 ms)
[ RUN      ] fourthLabTest.SinIntegralStaticTest3
[      OK  ] fourthLabTest.SinIntegralStaticTest3 (0 ms)
[ RUN      ] fourthLabTest.PrimeCountStaticTest
[      OK  ] fourthLabTest.PrimeCountStaticTest (0 ms)
[-----] 4 tests from fourthLabTest (0 ms total)
```

```

[-----] Global test environment tear-down
[=====] 4 tests from 1 test suite ran. (0 ms total)
[ PASSED ] 4 tests.
kristinab@LAPTOP-SFU9B1F4:~/ubuntu_main/OS_labs/build/tests$ ./lab4_test2
[=====] Running 4 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 4 tests from fourthLabTest
[ RUN      ] fourthLabTest.SinIntegralStaticTest1
[      OK   ] fourthLabTest.SinIntegralStaticTest1 (0 ms)
[ RUN      ] fourthLabTest.SinIntegralStaticTest2
[      OK   ] fourthLabTest.SinIntegralStaticTest2 (0 ms)
[ RUN      ] fourthLabTest.SinIntegralStaticTest3
[      OK   ] fourthLabTest.SinIntegralStaticTest3 (0 ms)
[ RUN      ] fourthLabTest.PrimeCountStaticTest
[      OK   ] fourthLabTest.PrimeCountStaticTest (0 ms)
[-----] 4 tests from fourthLabTest (0 ms total)

[-----] Global test environment tear-down
[=====] 4 tests from 1 test suite ran. (0 ms total)
[ PASSED ] 4 tests.

```

9 Выводы

В результате выполнения данной лабораторной работы были созданы динамические библиотеки, которые реализуют функционал в соответствии с вариантом задания на C++. Я приобрела практические навыки в создании программ, которые используют функции динамических библиотек.