

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: Слетюрин Кирилл Сергеевич

Группа: М8О-208Б-22

Преподаватель: Миронов Евгений Сергеевич

Вариант: 17

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 | Демонстрация работы программы | 9 |
| 8 | Запуск тестов | 9 |
| 9 | Выводы | 10 |

1 Репозиторий

<https://github.com/kirill483/OS>

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:

- Подсчёт количества простых чисел на отрезке
- Расчет значения числа Π при заданной длине ряда

В ходе выполнения лабораторной работы я использовала следующие системные вызовы:

- `dlopen` - открытие динамического объекта
- `dlclose` - закрытие динамического объекта
- `dlsym` - получить адрес, по которому символ из библиотеки расположен в памяти

5 Исходный код

lib1.cpp

```
1 #include "lib.hpp"
2
3 extern "C" int PrimeCount(int a, int b) {
4     int count = 0;
5     bool flag = true;
6     for (int i = a; i <= b; ++i) {
7         if (i <= 1) {
8             continue;
9         }
10        for (int j = 2; j < i; ++j) {
11            if (i % j == 0) {
12                flag = false;
13                break;
14            }
15        }
16        if (flag) {
17            ++count;
18        }
19        flag = true;
20    }
21    return count;
22 }
23
24 extern "C" void Swap(int* a, int* b) {
25     int temp = *a;
26     *a = *b;
27     *b = temp;
28 }
29
30 extern "C" int * Sort(int * array, int size) {
31     for (int i = 0; i < size - 1; i++) {
32         for (int j = 0; j < size - i - 1; j++) {
33             if (array[j] > array[j + 1]) {
34                 Swap(&array[j], &array[j + 1]);
35             }
36         }
37     }
38
39     return array;
40 }
```

lib2.cpp

```
1 #include "lib.hpp"
2
3 extern "C" int PrimeCount(int a, int b) {
4     int count = 0;
5     std::vector<int> numbers;
6     for (int i = 0; i <= b; ++i) {
7         numbers.push_back(i);
8     }
9     for (int i = 2; i <= b; ++i) {
10        if (numbers[i] != 0) {
11            if (numbers[i] >= a && numbers[i] <= b) {
12                ++count;
13            }
14            for (int j = i * i; j <= b; j += i) {
```

```

15         numbers[j] = 0;
16     }
17 }
18 }
19     return count;
20 }
21 //qsort
22
23
24 extern "C" void Swap(int* a, int* b) {
25     int temp = *a;
26     *a = *b;
27     *b = temp;
28 }
29
30 extern "C" int Partition(int * array, int low, int high) {
31     int pivot = array[(low + high) / 2];
32     int i = low;
33     int j = high;
34     while(true) {
35         while (array[i] < pivot) {
36             ++i;
37         }
38         while (array[j] > pivot) {
39             --j;
40         }
41         if (i >= j) {
42             return j;
43         }
44         Swap(&array[i++], &array[j--]);
45     }
46 }
47
48 extern "C" void Quicksort(int * array, int low, int high) {
49     if (low < high) {
50         int p = Partition(array, low, high);
51         Quicksort(array, low, p);
52         Quicksort(array, p + 1, high);
53     }
54 }
55
56 extern "C" int * Sort(int * array, int size) {
57     Quicksort(array, 0, size - 1);
58     return array;
59 }

```

staticmain.cpp

```

1 #include "lib.hpp"
2
3 #include <iostream>
4
5 void Task(const std::string& command) {
6     if (command == "1") {
7         int arg1;
8         std::cin >> arg1;
9         //int arg2[arg1];
10        int* arg2 = new int[arg1];
11        for (int i = 0; i < arg1; ++i) {
12            std::cin >> arg2[i];

```

```

13     }
14     Sort(arg2, arg1);
15     std::cout << "Result of sort = ";
16     for (int i = 0; i < arg1; ++i) {
17         std::cout << arg2[i] << " ";
18     }
19     std::cout << std::endl;
20 } else if (command == "2") {
21     int arg1, arg2;
22     std::cin >> arg1 >> arg2;
23     int result = PrimeCount(arg1, arg2);
24     std::cout << "Count of prime numbers = " << result << std
::endl;
25 } else {
26     std::cout << "Invalid command" << std::endl;
27 }
28 }
29
30 int main() {
31     std::string command;
32     while(true) {
33         std::cout << "Enter the command (0 - exit): ";
34         std::cin >> command;
35         if (command == "0") {
36             break;
37         }
38         Task(command);
39     }
40     return 0;
41 }

```

dynamicmain.cpp

```

1 #include "utils.hpp"
2
3 int main() {
4     const char pathToLib1[] = "/mnt/c/Users/Kirill/OS/lab4/build/
liblib1.so";
5     const char pathToLib2[] = "/mnt/c/Users/Kirill/OS/lab4/build/
liblib2.so";
6
7     void* libraryHandle = LoadLibrary(pathToLib1);
8     SortFunc Sort = (SortFunc)dlsym(libraryHandle, "Sort");
9     PrimeCountFunc PrimeCount = (PrimeCountFunc)dlsym(
libraryHandle, "PrimeCount");
10
11     std::string command;
12     while(true) {
13         std::cout << "Enter the command (0 - switch realization, e
- exit): ";
14         std::cin >> command;
15         if (command == "e") {
16             break;
17         } else if (command == "0") {
18             std::cout << "Enter the library (1 or 2): ";
19             std::cin >> command;
20             if (command == "1") {
21                 libraryHandle = LoadLibrary(pathToLib1);
22             } else if (command == "2") {
23                 libraryHandle = LoadLibrary(pathToLib2);

```

```

24         } else {
25             std::cout << "Invalid library" << std::endl;
26         }
27         Sort = (SortFunc)dlsym(libraryHandle, "Sort");
28         PrimeCount = (PrimeCountFunc)dlsym(libraryHandle, "
PrimeCount");
29     } else {
30         if (command == "1") {
31             std::cout << "Sort function:" << std::endl;
32             int arg1;
33             std::cin >> arg1;
34             //int arg2[arg1];
35             int* arg2 = new int[arg1];
36             for (int i = 0; i < arg1; ++i) {
37                 std::cin >> arg2[i];
38             }
39             Sort(arg2, arg1);
40             std::cout << "Result of sort = ";
41             for (int i = 0; i < arg1; ++i) {
42                 std::cout << arg2[i] << " ";
43             }
44             std::cout << std::endl;
45         } else if (command == "2") {
46             std::cout << "PrimeCount function:" << std::endl;
47             int arg1, arg2;
48             std::cin >> arg1 >> arg2;
49             int result = PrimeCount(arg1, arg2);
50             std::cout << "Count of prime numbers = " << result
<< std::endl;
51         } else {
52             std::cout << "Invalid command" << std::endl;
53         }
54     }
55 }
56 UnloadLibrary(libraryHandle);
57 return 0;
58 }

```

6 Тесты

```
1 #include "gtest/gtest.h"
2 #include "lib.hpp"
3
4 TEST(fourthLabTest, StaticTest1) {
5     int array[] = {0, 1, 2, 3, 4, 5};
6     int size = sizeof(array) / sizeof(int);
7     int * result = Sort(array, size);
8     for (int i = 0; i < size; ++i) {
9         EXPECT_EQ(result[i], i);
10    }
11 }
12
13 TEST(fourthLabTest, StaticTest2) {
14     int array[] = {5, 4, 3, 2, 1, 0};
15     int size = sizeof(array) / sizeof(int);
16     int * result = Sort(array, size);
17     for (int i = 0; i < size; ++i) {
18         EXPECT_EQ(result[i], i);
19    }
20 }
21
22 TEST(fourthLabTest, StaticTest3) {
23     int array[] = {3, 5, 2, 1, 4, 0};
24     int size = sizeof(array) / sizeof(int);
25     int * result = Sort(array, size);
26     for (int i = 0; i < size; ++i) {
27         EXPECT_EQ(result[i], i);
28    }
29 }
30
31 TEST(fourthLabTest, PrimeCountStaticTest1) {
32     int result = PrimeCount(3, 15);
33     EXPECT_EQ(result, 5);
34 }
35
36 TEST(fourthLabTest, PrimeCountStaticTest2) {
37     int result = PrimeCount(5, 125);
38     EXPECT_EQ(result, 28);
39 }
40
41 TEST(fourthLabTest, PrimeCountStaticTest3) {
42     int result = PrimeCount(16, 1634);
43     EXPECT_EQ(result, 252);
44 }
45
46 int main(int argc, char **argv) {
47     testing::InitGoogleTest(&argc, argv);
48     return RUN_ALL_TESTS();
49 }
```


7 Демонстрация работы программы

```
kirill@DESKTOP-7E05ERB:/mnt/c/Users/Kirill/OS/lab4/build$ ./static_main
Enter the command (0 - exit): 1
2 4
Result = 2
Enter the command (0 - exit): 0
```

8 Запуск тестов

```
kirill@DESKTOP-7E05ERB:/mnt/c/Users/Kirill/OS/lab4/build$ ./Test
[=====] Running 6 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 6 tests from fourthLabTest
[ RUN      ] fourthLabTest.StaticTest1
[      OK  ] fourthLabTest.StaticTest1 (0 ms)
[ RUN      ] fourthLabTest.StaticTest2
[      OK  ] fourthLabTest.StaticTest2 (0 ms)
[ RUN      ] fourthLabTest.StaticTest3
[      OK  ] fourthLabTest.StaticTest3 (0 ms)
[ RUN      ] fourthLabTest.PrimeCountStaticTest1
[      OK  ] fourthLabTest.PrimeCountStaticTest1 (0 ms)
[ RUN      ] fourthLabTest.PrimeCountStaticTest2
[      OK  ] fourthLabTest.PrimeCountStaticTest2 (0 ms)
[ RUN      ] fourthLabTest.PrimeCountStaticTest3
[      OK  ] fourthLabTest.PrimeCountStaticTest3 (1 ms)
[-----] 6 tests from fourthLabTest (1 ms total)

[-----] Global test environment tear-down
[=====] 6 tests from 1 test suite ran. (1 ms total)
[ PASSED  ] 6 tests.
```

9 Выводы

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