

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

Лабораторная работа №4 по курсу
«Операционные системы»

Студент: Кудрявов

Группа: М8О-208Б-22

Вариант: 21

Преподаватель: Миронов Евгений Сергеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2023

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/Marsha2022/OS.git>

Постановка задачи

Требуется создать динамические библиотеки, которые реализуют определенный функционал.

Далее использовать данные библиотеки 2-мя способами:

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

Общие сведения о программе

Программа состоит из одного заголовочного файла – lib.hpp и из двух его реализаций – lib1.cpp и lib2.cpp, которые будут линковаться разными способами в файлы static_main.cpp и dynamic_main.cpp соответственно.

Были использованы следующие системные вызовы:

1. dlopen()
2. dlsym()

Общий метод и алгоритм решения

В файлах lib1.cpp и lib2.cpp воплощены разные реализации функций с одним и тем же интерфейсом, которые по разному будут линковаться в файлах static_main.cpp и dynamic_main.cpp.

Исходный код

```
===== lib.hpp =====
```

```
#pragma once
```

```
#include <iostream>
```

```
#include <cmath>
```

```
#include <vector>
```

```
#include "utils.hpp"
```

```

#ifdef __cplusplus
extern "C" {
#endif

int PrimeCount(int a, int b);
int * Sort(int * array, int size);
void Swap(int* a, int* b);
int Partition(int * array, int low, int high);
void Quicksort(int * array, int low, int high);

#ifdef __cplusplus
}
#endif

===== utils.hpp =====

#include <iostream>
#include <cstdlib>
#include <dlfcn.h>

using SortFunc = int (*)(int *, int);
using PrimeCountFunc = int (*)(int, int);

void* LoadLibrary(const char *libraryName);
void UnloadLibrary(void* handle);

===== lib1.cpp =====

#include "lib.hpp"

extern "C" int PrimeCount(int a, int b) {
    int count = 0;
    bool flag = true;
    for (int i = a; i <= b; ++i) {
        if (i <= 1) {

```

```

        continue;
    }
    for (int j = 2; j < i; ++j) {
        if (i % j == 0) {
            flag = false;
            break;
        }
    }
    if (flag) {
        ++count;
    }
    flag = true;
}
return count;
}

extern "C" void Swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

//bubble
extern "C" int * Sort(int * array, int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (array[j] > array[j + 1]) {
                Swap(&array[j], &array[j + 1]);
            }
        }
    }

    return array;
}

```

===== lab2.cpp =====

```
#include "lib.hpp"
```

```
extern "C" int PrimeCount(int a, int b) {  
    int count = 0;  
    std::vector<int> numbers;  
    for (int i = 0; i <= b; ++i) {  
        numbers.push_back(i);  
    }  
    for (int i = 2; i <= b; ++i) {  
        if (numbers[i] != 0) {  
            if (numbers[i] >= a && numbers[i] <= b) {  
                ++count;  
            }  
            for (int j = i * i; j <= b; j += i) {  
                numbers[j] = 0;  
            }  
        }  
    }  
    return count;  
}  
//qsort
```

```
extern "C" void Swap(int* a, int* b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
extern "C" int Partition(int * array, int low, int high) {  
    int pivot = array[(low + high) / 2];  
    int i = low;
```

```

    int j = high;
    while(true) {
        while (array[i] < pivot) {
            ++i;
        }
        while (array[j] > pivot) {
            --j;
        }
        if (i >= j) {
            return j;
        }
        Swap(&array[i++], &array[j--]);
    }
}

extern "C" void Quicksort(int * array, int low, int high) {
    if (low < high) {
        int p = Partition(array, low, high);
        Quicksort(array, low, p);
        Quicksort(array, p + 1, high);
    }
}

extern "C" int * Sort(int * array, int size) {
    Quicksort(array, 0, size - 1);
    return array;
}

===== utils.cpp =====

#include "utils.hpp"

void* LoadLibrary(const char *libraryName) {
    void* handle = dlopen(libraryName, RTLD_LAZY);
    if (!handle) {

```

```

        std::cerr << "Couldn't load the library: " << dlerror() <<
std::endl;
        exit(EXIT_FAILURE);
    }
    return handle;
}

void UnloadLibrary(void* handle) {
    if (dlclose(handle) != 0) {
        std::cerr << "Couldn't unload the library: " << dlerror() <<
std::endl;
        exit(EXIT_FAILURE);
    }
}

===== dynamic_main.cpp =====
#include "utils.hpp"

int main() {
    const char pathToLib1[] =
"/home/hacker/prog/my_os_labs/build/lab4/liblib1.so";
    const char pathToLib2[] =
"/home/hacker/prog/my_os_labs/build/lab4/liblib2.so";

    void* libraryHandle = LoadLibrary(pathToLib1);
    SortFunc Sort = (SortFunc)dlsym(libraryHandle, "Sort");
    PrimeCountFunc PrimeCount = (PrimeCountFunc)dlsym(libraryHandle,
"PrimeCount");

    std::string command;
    while(true) {
        std::cout << "Enter the command (0 - switch realization, e -
exit): ";
        std::cin >> command;
        if (command == "e") {

```



```

        break;
    } else if (command == "0") {
        std::cout << "Enter the library (1 or 2): ";
        std::cin >> command;
        if (command == "1") {
            libraryHandle = LoadLibrary(pathToLib1);
        } else if (command == "2") {
            libraryHandle = LoadLibrary(pathToLib2);
        } else {
            std::cout << "Invalid library" << std::endl;
        }
        Sort = (SortFunc)dlsym(libraryHandle, "Sort");
        PrimeCount = (PrimeCountFunc)dlsym(libraryHandle,
"PrimeCount");
    } else {
        if (command == "1") {
            std::cout << "Sort function:" << std::endl;
            int arg1;
            std::cin >> arg1;
            //int arg2[arg1];
            int* arg2 = new int[arg1];
            for (int i = 0; i < arg1; ++i) {
                std::cin >> arg2[i];
            }
            Sort(arg2, arg1);
            std::cout << "Result of sort = ";
            for (int i = 0; i < arg1; ++i) {
                std::cout << arg2[i] << " ";
            }
            std::cout << std::endl;
        } else if (command == "2") {
            std::cout << "PrimeCount function:" << std::endl;
            int arg1, arg2;
            std::cin >> arg1 >> arg2;

```

```

        int result = PrimeCount(arg1, arg2);
        std::cout << "Count of prime numbers = " << result <<
std::endl;
    } else {
        std::cout << "Invalid command" << std::endl;
    }
}
}
}
UnloadLibrary(libraryHandle);
return 0;
}

```

===== static_main.cpp =====

```
#include "lib.hpp"
```

```
#include <iostream>
```

```

void Task(const std::string& command) {
    if (command == "1") {
        int arg1;
        std::cin >> arg1;
        int* arg2 = new int[arg1];
        for (int i = 0; i < arg1; ++i) {
            std::cin >> arg2[i];
        }
        Sort(arg2, arg1);
        std::cout << "Result of sort = ";
        for (int i = 0; i < arg1; ++i) {
            std::cout << arg2[i] << " ";
        }
        std::cout << std::endl;
    } else if (command == "2") {
        int arg1, arg2;
        std::cin >> arg1 >> arg2;
        int result = PrimeCount(arg1, arg2);
    }
}

```

```

        std::cout << "Count of prime numbers = " << result <<
std::endl;
    } else {
        std::cout << "Invalid command" << std::endl;
    }
}

int main() {
    std::string command;
    while(true) {
        std::cout << "Enter the command (0 - exit): ";
        std::cin >> command;
        if (command == "0") {
            break;
        }
        Task(command);
    }
    return 0;
}

```

Демонстрация работы программы

marshall@marshall:~/Desktop/OS_labs/build/lab4\$./dynamic_m

ai n

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

Sort function:

6

3 4 5 2 5 3

Result of sort = 2 3 3 4 5 5

Enter the command (0 - switch realization, e - exit): 2

PrimeCount function:

3 150

Count of prime numbers = 34

Enter the command (0 - switch realization, e - exit): 0

Enter the library (1 or 2): 2

Enter the command (0 - switch realization, e - exit): 6

Invalid command

Enter the command (0 - switch realization, e - exit): 2

PrimeCount function:

3 150

Count of prime numbers = 34

Enter the command (0 - switch realization, e - exit): 1

Sort function:

6

3 4 5 2 5 3

Result of sort = 2 3 3 4 5 5

Enter the command (0 - switch realization, e - exit): e

Выводы

В ходе выполнения лабораторной работы были получены глубокие знания о процессе сборки и компиляции программ на языке Си и C++. Была изучена разница между динамической линковкой на этапе компиляции и на этапе выполнения программы.