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: Друхольский А.К.

Группа: М8О-207Б-21

Вариант: 20

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/ssForz/OS-labs

**Постановка задачи**

Целью является приобретение практических навыков в:

* Создание динамических библиотек
* Создание программ, которые используют функции динамических библиотек
* Работа со сборочной системой

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 используют 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 обоих 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;
3. «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.

Так же нужно сделать файл cmake с особенностями согласно выданному варианту

**Общие сведения о программе**

У нас имеется два файла lib1.cpp и lib2.cpp, в каждом из которых представлена одна из реализаций выданной функции. config.h и config.h.in — файлы, служащие для отображения данных компиляции (вариант cmake). Main1.cpp — файл, которому предписываются библиотеки на этапе компиляции. Из него получается два исполняемых файла main1 и main2. Main2.cpp — файл, использующий динамические библиотеки. Из него получается исполняемый файл main. Lib.h нужен для подключения библиотек.

**Общий метод и алгоритм решения**

В одной реализации: перевод в двоичную систему и вывод простых чисел примитивным алгоритмом

В другой реализации: перевод в троичную систему и вывод простых чисел с помощью решета эратосфена

Реализуем три исполняемых файла. Первые два с библиотеками, подключенными на этапе линковки. Последний будет способен переключать реализации.

**Исходный код**

**lib.h**

#ifndef \_\_LIB\_H\_\_

#define \_\_LIB\_H\_\_

extern "C" int PrimeCount(int a, int b);

extern "C" char\* translation(long x);

#endif

**lib1.cpp**

#include<iostream>

using namespace std;

extern "C" int PrimeCount(int a, int b)

{

if (a > b) {

int c = b;

b = a;

a = c;

}

int counter = 0;

for (int i = a; i <= b; i++) {

int flag = 0;

for (int k = 2; k < i; k++) {

if (i%k == 0) {

flag = 1;

break;

}

}

if (flag == 0) {

counter++;

}

}

return counter;

}

extern "C" char\* translation(long x)

{

if (x == -1) {

cout<<"binary";

}

int cnt = 0;

int sizelong = 31;

char\* binary = new char[sizelong];

for (int i = 0; i < sizelong; i++) {

binary[i] = '9';

}

while(x > 0) {

if (x%2 == 1) {

binary[sizelong - cnt - 1] = '1';

} else {

binary[sizelong - cnt - 1] = '0';

}

x = x/2;

cnt++;

}

return binary;

}

/\*int main()

{

int a, b;

long p;

char\* answ;

cin>>a>>b;

cin>>p;

cout<<PrimeCount(a,b)<<endl;

answ = translation(p);

for (int i = 0; i < 32; i++) {

if (answ[i] == '1' || answ[i] == '0') {

cout << answ[i];

}

}

cout<<endl;

delete answ;

return 0;

}\*/

**lib2.cpp**

#include<iostream>

using namespace std;

extern "C" int PrimeCount(int a, int b)

{

if (a > b) {

int c = b;

b = a;

a = c;

}

int counter = 0;

int resheto[b];

for (int i = 0; i < b + 1; i++) {

resheto[i] = i;

}

for (int p = 2; p < b + 1; p++) {

if (resheto[p] != 0) {

for (int j = p\*p; j < b + 1; j+=p) {

resheto[j] = 0;

}

}

}

for (int i = a; i <= b; i++) {

if (resheto[i] != 0) {

counter++;

}

}

return counter;

}

extern "C" char\* translation(long x)

{

if (x == -1) {

cout<<"ternary";

}

int cnt = 0;

int sizelong = 20;

char\* ternary = new char[sizelong];

for (int i = 0; i < sizelong; i++) {

ternary[i] = '9';

}

while(x > 0) {

if (x%3 == 1) {

ternary[sizelong - cnt - 1] = '1';

} else if (x%3 == 2){

ternary[sizelong - cnt - 1] = '2';

} else {

ternary[sizelong - cnt - 1] = '0';

}

x = x/3;

cnt++;

}

return ternary;

}

/\*int main()

{

int a, b;

long p;

char\* answ;

cin>>a>>b;

cin>>p;

cout<<PrimeCount(a,b)<<endl;

answ = translation(p);

for (int i = 0; i < 20; i++) {

if (answ[i] == '1' || answ[i] == '0' || answ[i] == '2') {

cout << answ[i];

}

}

cout<<endl;

delete answ;

return 0;

}\*/

**main1.cpp**

#include<iostream>

#include<stdio.h>

#include<string>

#include"lib.h"

#include"config.h"

using namespace std;

int main()

{

cout<<"Compiler version: "<<COMP\_VER<<endl;

cout<<"Compiler ID: "<<COMP\_ID<<endl;

cout<<"Compilation date: "<<TIME\_NOW<<endl;

cout<<"You should write commands: <command> <arg1> <arg2> ... <argn>"<<endl;

cout<<"If you want to count prime numbers in range [a,b], write: 1 <a> <b>"<<endl;

cout<<"If you want to translate number from decimal to ";

translation(-1);

cout<<", write: 2 <number>"<<endl;

int command = 0;

while(cin>>command) {

if (command == 1) {

int a, b;

int cnt;

cin>>a>>b;

cnt = PrimeCount(a, b);

cout<<"Count of rime numbers in range ["<<a<<","<<b<<"]: "<<cnt<<endl;

continue;

}

if (command == 2) {

long a;

char\* answ;

cin>>a;

answ = translation(a);

cout<<"Result of translation:"<<endl;

for (int i = 0; i < 32; i++) {

if (answ[i] == '1' || answ[i] == '0' || answ[i] == '2') {

cout << answ[i];

}

}

cout<<endl;

delete answ;

continue;

}

cout<<"Wrong command input"<<endl;

}

}

**main2.cpp**

#include<iostream>

#include<stdio.h>

#include<dlfcn.h>

#include"lib.h"

#include<string>

#include"config.h"

using namespace std;

int main()

{

cout<<"Compiler version: "<<COMP\_VER<<endl;

cout<<"Compiler ID: "<<COMP\_ID<<endl;

cout<<"Compilation date: "<<TIME\_NOW<<endl;

string lib1 = "./liblib1.so";

string lib2 = "./liblib2.so";

int command;

cout<<"You are in program-1 now"<<endl;

cout<<"You should write commands: <command> <arg1> <arg2> ... <argn>"<<endl;

cout<<"If you want to change realization of calculation, write 0"<<endl;

cout<<"If you want to count prime numbers in range [a,b], write: 1 <a> <b>"<<endl;

cout<<"If you want to translate number from decimal to other, write: 2 <number>"<<endl;

void\* curlib = dlopen(lib1.c\_str(), RTLD\_LAZY);

cout<<"Using lib1 at start"<<endl;

int (\*PrimeCount)(int a, int b);

char\* (\*translation)(long x);

PrimeCount = (int(\*)(int, int))dlsym(curlib, "PrimeCount");

translation = (char\*(\*)(long))dlsym(curlib, "translation");

//CountPrime = dlsym(curlib, "CountPrime");

//translation = dlsym(curlib, "translation");

int lib\_id = 1;

while(cin>>command) {

if (command == 0) {

dlclose(curlib);

if (lib\_id == 1) {

curlib = dlopen(lib2.c\_str(), RTLD\_LAZY);

lib\_id = 2;

} else {

curlib = dlopen(lib1.c\_str(), RTLD\_LAZY);

lib\_id= 1;

}

PrimeCount = (int(\*)(int, int))dlsym(curlib, "PrimeCount");

translation = (char\*(\*)(long))dlsym(curlib, "translation");

continue;

}

if (command == 1) {

int a, b;

int cnt;

cin>>a>>b;

cnt = PrimeCount(a, b);

cout<<"Count of prime numbers in range ["<<a<<","<<b<<"]: "<<cnt<<endl;

continue;

}

if (command == 2) {

long a;

char\* answ;

cin>>a;

answ = translation(a);

cout<<"Result of translation to ";

translation(-1);

cout<<":"<<endl;

for (int i = 0; i < 32; i++) {

if (answ[i] == '1' || answ[i] == '0' || answ[i] == '2') {

cout << answ[i];

}

}

cout<<endl;

delete answ;

continue;

}

cout<<"Wrong command input"<<endl;

}

}

**config.h.in**

#ifndef CONFIG\_H\_IN

#define CONFIG\_H\_IN

#define PROJECT\_NAME "@PROJECT\_NAME@"

#define COMP\_ID "@COMP\_ID@"

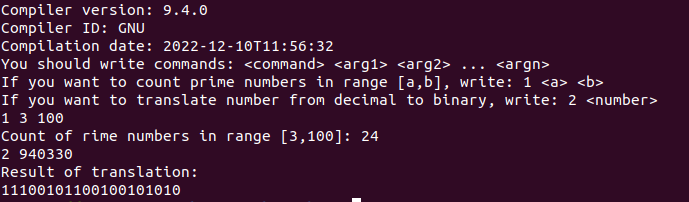
#define COMP\_VER "@COMP\_VER@"

#define TIME\_NOW "@TIME\_NOW@"

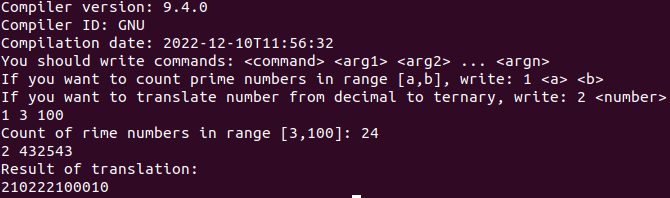
#endif // CONFIG\_H\_IN

**Демонстрация работы программы**

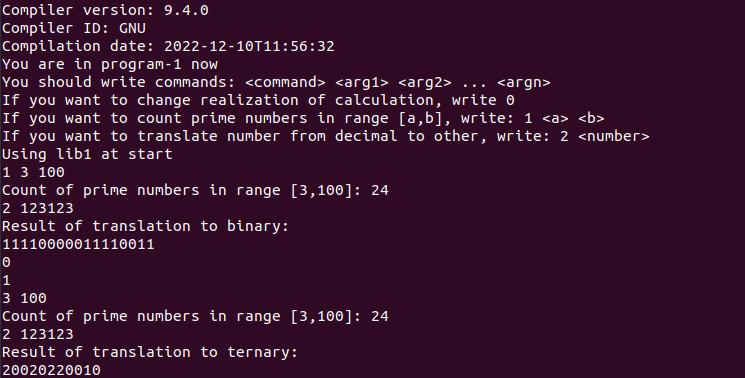
main1:

****

main2:



main:



**Выводы**

В данной лабораторной работе я познакомился с динамическими библиотеками. Так же мне удалось углубиться в cmake и почувствовать его возможности, но в то же время я столкнулся и с проблемами. Я понял, что большое количество информации доступно только на другом языке, поэтому следует изначально искать решения проблем на англоязычных сайтах.