**Московский Авиационный Институт**

**(Национальный Исследовательский Университет)**

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №4 по курсу**

**«Операционные системы»**

Группа: М80-203Б-22

Студент: Касумова Н.Р.

Преподаватель: Миронов Е.С.

Оценка:

Подпись:

Дата: 27.12.23

**Цель работы:**

Приобретение практических навыков в:

* Создание динамических библиотек
* Создание программ, которые используют функции динамических библиотек

**Постановка задачи**

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).

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.

Вариант 18:

* Подсчет числа простых элементов на заданном промежутке
  + Перебор
  + Решето Эратосфена
* Подсчет числа E (натуральный логарифм)
  + С помощью формулы (1 + 1 / x)^x
  + Сумма рядов по n от 0 до x, вычисляя по формуле (1 / n!)

Сборка и запуск:

gcc -fPIC -c lib1.c -o d1.o

gcc -fPIC -c lib2.c -o d2.o

gcc -shared -o libd1.so d1.o -lm

gcc -shared -o libd2.so d2.o -lm

gcc prog1.c -L -ld1 -o main1 -WI, -rpath -WI, -lm

gcc prog2.c -L -ld2 -o main2 -WI, -rpath -WI, -lm

Флаг –fPIC означает, что используется относительная переадресация в переходах подпрограмм.

Флаг –shared предписывается, что создается динамическая библиотека.

Флаг –L. - ищет в каталоге файлы библиотеки.

Флаг –ld<> используется для работ с функциями dlopen, dlsym, dlclose.

Флаг –lm/-ld1 связывает с файлом библиотеку math/lib1.

**Код программы**

Lib01.c:

#include <math.h>

int IsPrime(int n) {

if(n < 2) {

return 0;

}

for(int i = 2; i \* i <= n; i++) {

if(n % i == 0) {

return 0;

}

}

return 1;

}

int PrimeCount(int a, int b) {

int count = 0;

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

if(IsPrime(i) == 1) {

count++;

}

}

return count;

}

float E(int x) {

return pow((1 + 1. / x), x);

}

Lib02.c

#include <stdlib.h>

#include <math.h>

int PrimeCount(int a, int b) {

int \*primes = (int\* )malloc((b + 1) \* sizeof(int));

int count = 0;

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

primes[i] = 1;

}

primes[0] = primes[1] = 0;

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

if(primes[p] == 1) {

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

primes[i] = 0;

}

}

}

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

if(primes[i] == 1) {

count++;

}

}

free(primes);

return count;

}

float E(int x) {

double ans = 0, s = 1;

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

if(i != 0) {

s \*= i;

}

ans += 1. / s;

}

return ans;

}

Prog1.c

#include <stdio.h>

int PrimeCount(int a,int b);

float E(int x);

int main() {

int key;

while(scanf("%d", &key) > 0) {

if(key == 1) {

int a, b;

scanf("%d%d", &a, &b);

printf("Prime count on segment from %d to %d : %d\n", a, b, PrimeCount(a, b));

}

else if(key == 2) {

int x;

scanf("%d", &x);

printf("E(%d) = %f\n", x, E(x));

}

else {

printf("Error\n");

}

}

return 0;

}

Prog2.c

#include <stdio.h>

#include <dlfcn.h>

int main(int argc, char\* argv[]) {

void \*ext\_library = NULL;

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

float (\*E)(int x);

int key, start\_library;

printf("Choose start library: 1 or 2: ");

scanf("%d\n", &start\_library);

if(start\_library == 1) {

ext\_library = dlopen("/Users/n/Projects/OS/lab\_4/libd1.so", RTLD\_LAZY);

}

else if(start\_library == 2) {

ext\_library = dlopen("/Users/n/Projects/OS/lab\_4/libd2.so", RTLD\_LAZY);

}

else {

printf("Don't write right key\n");

}

if(!ext\_library) {

fprintf(stderr, "dlopen() error %s\n", (char\* )dlerror);

return 1;

}

PrimeCount = dlsym(ext\_library, "PrimeCount");

E = dlsym(ext\_library, "E");

while (scanf("%d",&key) > 0) {

if(key == 1) {

int a, b;

scanf("%d%d", &a, &b);

printf("%s(%d, %d) = %d\n","PrimeCount", a, b, (\*PrimeCount)(a, b));

}

else if(key == 2) {

int x;

scanf("%d", &x);

printf("%s(%d) = %f\n","E",x,(\*E)(x));

}

else if(key == 0) {

dlclose(ext\_library);

if(start\_library == 1) {

ext\_library = dlopen("./libd2.so", RTLD\_LAZY);

start\_library = 2;

}

else if(start\_library == 2) {

ext\_library = dlopen("./libd1.so", RTLD\_LAZY);

start\_library = 1;

}

if(!ext\_library) {

fprintf(stderr,"dlopen() error: %s\n", (char\* )dlerror());

return 1;

}

PrimeCount = dlsym(ext\_library,"PrimeCount");

E = dlsym(ext\_library, "E");

}

}

dlclose(ext\_library);

}

**Примеры работы**

n@MacBook-Pro-N lab\_4 % ./main1

1 1 17

Prime count on segment from 1 to 17 : 7

2 1

E(1) = 2.000000

2 10

E(10) = 2.593742

2 100

E(100) = 2.704814

8

Error

n@MacBook-Pro-N lab\_4 % ./main2

Choose start library: 1 or 2: 1

2 1

E(1) = 2.000000

2 2

E(2) = 2.250000

2 3

E(3) = 2.370370

2 4

E(4) = 2.441406

1 1 5

PrimeCount(1, 5) = 3

1 2 3

PrimeCount(2, 3) = 2

0

2 1

E(1) = 2.000000

2 2

E(2) = 2.500000

2 5

E(5) = 2.716667

2 100

E(100) = 2.718282

2 3

E(3) = 2.666667

1 1 5

PrimeCount(1, 5) = 3

**Strace**

n@MacBook-Pro-N lab\_4 % ./main2

Choose start library: 1 or 2: 1

1

5 8

PrimeCount(5, 8) = 2

2

5

E(5) = 2.488320

0

2

5

E(5) = 2.716667

1

3 150

PrimeCount(3, 150) = 34

**Вывод**

В ходе выполнения лабораторной работы я узнала как работать с динамическими библиотеками. Если говорить конкретней, то процесс их создания, «подгружение» на этапе компиляции и на этапе работе программы. Мне показалось интересным опытом данная лабораторная, так как можно с легкостью расширять код, добавляя туда новые функции.