

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: М8О-209БВ-24

Студент: Осипов М.Н.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 14.12.25

Москва, 2025

Постановка задачи

Вариант 18.

Функция 1: Описание: подсчёт количества простых чисел на отрезке $[A, B]$ (A, B - натуральные). Сигнатура: `int PrimeCount(int A, int B)`. Реализация 1: наивный алгоритм. Проверить делимость текущего числа на все предыдущие числа. Реализация 2: решето Эратосфена.

Функция 2: Описание: расчет значения числа e (основание натурального логарифма). Сигнатура: `Float E(int x)`. Реализация 1: $(1 + 1/x)^x$. Реализация 2: сумма ряда по n от 0 до x , где элементы ряда равны: $(1/(n!))$

Общий метод и алгоритм решения

Использованные системные вызовы:

- `dlopen()/dlclose()` - загрузка библиотеки в память, открытие файла `.so`/выгрузка библиотеки из памяти, закрытие файла библиотеки
- `scanf()` - чтение пользовательского ввода
- `printf()` - вывод результатов
- `malloc()/free()` - Управление кучей
- `return` - Завершение программы

Алгоритм работы:

Программа представляет собой интерактивное консольное приложение, которое позволяет пользователю выполнять две математические операции:

подсчёт количества простых чисел на заданном отрезке $[A, B]$ (где A и B — натуральные числа);

вычисление приближённого значения числа e (основания натурального логарифма) при заданном параметре x .

В динамическом режиме программа также поддерживает переключение между двумя различными реализациями этих функций, выполненными в отдельных динамических библиотеках.

Программа работает в бесконечном цикле, пока пользователь не введёт команду для выхода. Ниже приведено пошаговое описание работы.

Режимы работы:

Статическая линковка (`static`) — функции `PrimeCount` и `E` подключены на этапе линковки;

Динамическая загрузка (`dynamic`) — функции загружаются из библиотек при помощи `dlopen()` и `dlsym()`.

Код программы

mathlib.h

```
#ifndef MATHLIB_H

#define MATHLIB_H


int PrimeCount(int A, int B);

float E(int x);


#endif
```

dynamic.c

```
#include <stdio.h>

#include <stdlib.h>

#include <dlfcn.h>

#include "mathlib.h"


int main() {

    char *libs[] = {"/lib1.so", "/lib2.so"};

    int cur = 0;


    void *handle = dlopen(libs[cur], RTLD_LAZY);

    if (!handle) {

        printf("Ошибка загрузки библиотеки: %s\n", dlerror());

        return 1;

    }


    int (*PrimeCountFunc)(int, int) = dlsym(handle, "PrimeCount");

    float (*EFunc)(int) = dlsym(handle, "E");


    if (!PrimeCountFunc || !EFunc) {

        printf("Ошибка загрузки символов: %s\n", dlerror());
```

```
    dlclose(handle);

    return 1;
}

int cmd;
int arg1, arg2;

while (1) {
    if (scanf("%d", &cmd) == EOF) {
        printf("ВЫХОД.\n");
        break;
    }

    if (cmd == -1) {
        printf("ВЫХОД.\n");
        break;
    }

    if (cmd == 0) {
        dlclose(handle);
        cur = 1 - cur;

        handle = dlopen(libs[cur], RTLD_LAZY);
        if (!handle) {
            printf("Ошибка загрузки библиотеки: %s\n", dlerror());
            return 1;
        }
    }
}
```

```
PrimeCountFunc = dlsym(handle, "PrimeCount");

EFunc = dlsym(handle, "E");


if (!PrimeCountFunc || !EFunc) {

    printf("Ошибка загрузки символов: %s\n", dlerror());

    dlclose(handle);

    return 1;

}


printf("Библиотека переключена на: %s\n", libs[cur]);

}

else if (cmd == 1) {

    if (scanf("%d %d", &arg1, &arg2) != 2) {

        printf("Ошибка: для команды 1 требуется два аргумента A и B.\n");

        continue;

    }

    printf("PrimeCount(%d, %d) = %d\n", arg1, arg2, PrimeCountFunc(arg1, arg2));

}

else if (cmd == 2) {

    if (scanf("%d", &arg1) != 1) {

        printf("Ошибка: для команды 2 требуется один аргумент x.\n");

        continue;

    }

    printf("E(%d) = %f\n", arg1, EFunc(arg1));

}

else {

    printf("Нет такой команды. Используйте 1, 2, 0 или -1 для выхода.\n");

}

}
```

```
    dlclose(handle);

    return 0;

}
```

lib1.c

```
#include <math.h>

#include "mathlib.h"

int PrimeCount(int A, int B) {

    int count = 0;

    for (int i = A; i <= B; i++) {

        if (i < 2) continue;

        int is_prime = 1;

        for (int j = 2; j <= i / 2; j++) {

            if (i % j == 0) {

                is_prime = 0;

                break;

            }

        }

        if (is_prime) {

            count++;

        }

    }

    return count;

}

float E(int x) {

    if (x <= 0) return 1.0f;

    return powf(1.0f + 1.0f / (float)x, (float)x);

}
```

lib2.c

```
#include <stdlib.h>

#include <string.h>

#include "mathlib.h"

int PrimeCount(int A, int B) {

    if (B < 2 || A > B) return 0;

    if (A < 2) A = 2;

    char *is_prime = (char*)malloc((B + 1) * sizeof(char));

    if (!is_prime) return -1;

    memset(is_prime, 1, (B + 1) * sizeof(char));

    is_prime[0] = is_prime[1] = 0;

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

        if (is_prime[i]) {

            for (int j = i * i; j <= B; j += i) {

                is_prime[j] = 0;

            }

        }

    }

    int count = 0;

    for (int i = A; i <= B; i++) {

        if (is_prime[i]) count++;

    }

    free(is_prime);
```

```
    return count;
}

float E(int x) {
    if (x < 0) return 1.0f;

    float e = 1.0f;
    float factorial = 1.0f;
    for (int n = 1; n <= x; n++) {
        factorial *= n;
        e += 1.0f / factorial;
    }
    return e;
}
```

static.c

```
#include <stdio.h>

#include "mathlib.h"

int main() {
    int cmd;
    int arg1, arg2;

    while (1) {
        if (scanf("%d", &cmd) == EOF) {
            printf("Выход.\n");
            break;
        }

        if (cmd == -1) {
```



```

        printf("Выход.\n");
        break;
    }

    if (cmd == 1) {
        if (scanf("%d %d", &arg1, &arg2) != 2) {
            printf("Ошибка: для команды 1 требуется два аргумента А и В.\n");
            continue;
        }
        printf("PrimeCount(%d, %d) = %d\n", arg1, arg2, PrimeCount(arg1, arg2));
    }
    else if (cmd == 2) {
        if (scanf("%d", &arg1) != 1) {
            printf("Ошибка: для команды 2 требуется один аргумент х.\n");
            continue;
        }
        printf("E(%d) = %f\n", arg1, E(arg1));
    }
    else if (cmd == 0) {
        printf("В статической версии переключение библиотек недоступно.\n");
    }
    else {
        printf("Нет такой команды. Используйте 1, 2, 0 или -1 для выхода.\n");
    }
}

return 0;
}

```

Протокол работы программы

./dynamic_program

1 10 30

PrimeCount(10, 30) = 6

2 5

E(5) = 2.488321

0

Библиотека переключена на: ./lib2.so

1 1 20

PrimeCount(1, 20) = 8

-1

Выход.

strace -f ./dynamic_program

execve("./dynamic_program", ["./dynamic_program"], 0x7ffcd6b58838 /* 27 vars */) = 0

brk(NULL) = 0x60f99e756000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc027af860) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x78f2a5843000

access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=61360, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 61360, PROT_READ, MAP_PRIVATE, 3, 0) = 0x78f2a5834000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0@\0\0\0\0\0\0@\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0"..., 48, 848) = 48

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0O{\f225\\=\201\327\312\301P\32\$\230\266\235"..., 68, 896) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0@\0\0\0\0\0\0@\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x78f2a5600000

mprotect(0x78f2a5628000, 2023424, PROT_NONE) = 0

mmap(0x78f2a5628000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x78f2a5628000

mmap(0x78f2a57bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x78f2a57bd000

mmap(0x78f2a5816000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x78f2a5816000

mmap(0x78f2a581c000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x78f2a581c000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x78f2a5831000

arch_prctl(ARCH_SET_FS, 0x78f2a5831740) = 0

set_tid_address(0x78f2a5831a10) = 1918

set_robust_list(0x78f2a5831a20, 24) = 0

rseq(0x78f2a58320e0, 0x20, 0, 0x53053053) = 0

mprotect(0x78f2a5816000, 16384, PROT_READ) = 0

mprotect(0x60f964571000, 4096, PROT_READ) = 0

mprotect(0x78f2a587d000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x78f2a5834000, 61360) = 0

getrandom("\x53\x30\xb7\x42\xaa\x91\x66\xcb", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x60f99e756000

brk(0x60f99e777000) = 0x60f99e777000

openat(AT_FDCWD, "./lib1.so", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15584, ...}, AT_EMPTY_PATH) = 0

getcwd("/mnt/c/Users/\320\237\320\276\320\273\321\214\320\267\320\276\320\262\320\260\321\202\320\265\320\273\321\214\OneDrive\Desktop\2 \320\272\321\203\321\200\321\201\Laby_OC\Lab_4/src", 128) = 84

mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x78f2a583e000

mmap(0x78f2a583f000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x78f2a583f000

mmap(0x78f2a5840000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x78f2a5840000

mmap(0x78f2a5841000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x78f2a5841000

close(3) = 0

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=61360, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 61360, PROT_READ, MAP_PRIVATE, 3, 0) = 0x78f2a55f1000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x78f2a550a000

mmap(0x78f2a5518000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x78f2a5518000

mmap(0x78f2a5594000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x78f2a5594000

mmap(0x78f2a55ef000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x78f2a55ef000

close(3) = 0

mprotect(0x78f2a55ef000, 4096, PROT_READ) = 0

mprotect(0x78f2a5841000, 4096, PROT_READ) = 0

munmap(0x78f2a55f1000, 61360) = 0

newfstatat(0, "", {st_mode=S_IFCHR|0600, st_rdev=makedev(0x88, 0x2), ...}, AT_EMPTY_PATH) = 0

read(0, 1

"1\n", 1024) = 2

read(0, 10

"10\n", 1024) = 3

read(0, 30

"30\n", 1024) = 3

newfstatat(1, "", {st_mode=S_IFCHR|0600, st_rdev=makedev(0x88, 0x2), ...}, AT_EMPTY_PATH) = 0

write(1, "PrimeCount(10, 30) = 6\n", 23PrimeCount(10, 30) = 6

) = 23

read(0, 2

```

"2\n", 1024)          = 2

read(0, 5

"5\n", 1024)          = 2

write(1, "E(5) = 2.488321\n", 16E(5) = 2.488321

)      = 16

read(0, 0

"0\n", 1024)          = 2

munmap(0x78f2a583e000, 16432)      = 0

munmap(0x78f2a550a000, 942344)      = 0

openat(AT_FDCWD, "./lib2.so", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15704, ...}, AT_EMPTY_PATH) = 0

getcwd("/mnt/c/Users/

\320\237\320\276\320\273\321\214\320\267\320\276\320\262\320\260\321\202\320\265\320\273\321\21

4/OneDrive/Desktop/2 \320\272\321\203\321\200\321\201/Laby_OC/Lab_4/src", 128) = 84

mmap(NULL, 16448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =

0x78f2a583e000

mmap(0x78f2a583f000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|

MAP_DENYWRITE, 3, 0x1000) = 0x78f2a583f000

mmap(0x78f2a5840000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|

MAP_DENYWRITE, 3, 0x2000) = 0x78f2a5840000

mmap(0x78f2a5841000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|

MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x78f2a5841000

close(3)              = 0

mprotect(0x78f2a5841000, 4096, PROT_READ) = 0

write(1, "\320\221\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320\272\320\260

\320\277\320\265\321\200\320\265\320\272\320"..., 60Библиотека переключена на: ./lib2.so

) = 60

read(0, 1

"1\n", 1024)          = 2

read(0, 1

"1\n", 1024)          = 2

read(0, 20

"20\n", 1024)         = 3

write(1, "PrimeCount(1, 20) = 8\n", 22PrimeCount(1, 20) = 8

```

) = 22

read(0, -1

"-1\n", 1024) = 3

write(1, "\320\222\321\213\321\205\320\276\320\264.\n", 12Выход.

) = 12

munmap(0x78f2a583e000, 16448) = 0

lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)

exit_group(0) = ?

+++ exited with 0 +++

Вывод

В ходе выполнения лабораторной работы номер 4 я составил и отладил программу на языке Си с реализацией двух механизмов работы с библиотеками с возможностью переключаться между реализациями без перекомпиляции: статическим и динамическим, были решены 2 различные математические функции.