

# Отчет по лабораторной работе № 5 по курсу Операционные системы

Студент группы М8О-206Б-21 Синюков Антон Сергеевич, № по списку 19

Контакты www, e-mail, icq, skype vk.com/antonckya

Работа выполнена: « 27 » декабря 2022 г.

Преподаватель: Миронов Евгений

Входной контроль знаний с оценкой \_\_\_\_\_

Отчет сдан «    » \_\_\_\_\_ 202 \_\_ г., итоговая оценка \_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Динамические библиотеки  
\_\_\_\_\_
2. **Цель работы:** Приобретение практических навыков в создании динамических библиотек и создании программ, которые используют функции динамических библиотек  
\_\_\_\_\_  
\_\_\_\_\_
3. **Задание (вариант № 9):** 1. Расчет производной функции  $\cos(x)$  в точке А с приращением  $\delta x$ .  
2. Подсчёт количества простых чисел на отрезке  $[A, B]$  (А, В - натуральные)  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
4. **Оборудование:**  
ЭВМ \_\_\_\_\_, процессор \_\_\_\_\_, имя узла сети \_\_\_\_\_ с ОП \_\_\_\_\_ Мб,  
НМД \_\_\_\_\_ Мб. Терминал \_\_\_\_\_ адрес \_\_\_\_\_. Принтер \_\_\_\_\_  
Другие устройства \_\_\_\_\_  
\_\_\_\_\_
5. **Программное обеспечение:**  
Операционная система семейства \_\_\_\_\_, наименование \_\_\_\_\_ версия \_\_\_\_\_  
интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_  
Система программирования \_\_\_\_\_ версия \_\_\_\_\_  
Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_  
Утилиты операционной системы \_\_\_\_\_  
\_\_\_\_\_  
Прикладные системы и программы \_\_\_\_\_  
Местонахождение и имена файлов программ и данных \_\_\_\_\_  
\_\_\_\_\_
6. **Идея, метод, алгоритм** решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)  
  
Библиотеки находятся в lib1.c и lib2.c, названия у функций одинаковые, так что у них общий header lab.h.  
Подгрузка и динамическая смена библиотеке осуществляется в main.c.  
Прик компиляции кода lib1.c и lib2.c компилируются в .so файлы. Подгрузка библиотек программой осуществляется с помощью dlopen, определение соответствующих функций с помощью dlsym.
7. **Сценарий выполнения работы** (план работы, первоначальный текст программы в черновике [можно на отдельном листе] и тесты либо соображения по тестированию)

1. Изучить работу с динамическими библиотеками: создание и их использование в ОС Linux с помощью `dlfcn.h`
2. Сделать `lib1.c` и `lib2.c`
3. Написать `main1.c` - программа, которая загружает библиотеки статически
4. Написать `main2.c` - программа, которая загружает библиотеки динамически
5. Скомпилировать и протестировать программу.

**8. Выводы:** Эта лабораторная работа оказалась довольно интересной, мы познакомились с работой многих алгоритмов аллокации и распределения динамической памяти на лекциях, познакомились с написанием динамических библиотек на Си. Такие библиотеки возможно использовать не только в языке Си, но и в более удобных, но медленных языках.

---

---

---

---

---

---

---

---

---

---

Подпись студента \_\_\_\_\_

## 9. Код

### lib.h

```
#ifndef LAB5_LIB_H
#define LAB5_LIB_H

float Derivative(float A, float deltaX);

int PrimeCount(int A, int B);

#endif
```

### lib1.c

```
#include <math.h>

float Derivative(float A, float deltaX){
    return (cos(A + deltaX) - cos(A)) / deltaX;
}

int PrimeCount(int A, int B){
    int prime_count = 0;
    for (int i = A; i <= B; i++){
        int is_prime = 1;
        for (int j = 2; j < i; j++){
            if (i % j == 0)
                is_prime = 0;
        }
        if (i < 2)
            is_prime = 0;
        prime_count += is_prime;
    }
    return prime_count;
}
```

### lib2.c

```
#include <math.h>
#include <stdlib.h>

float Derivative(float A, float deltaX){
    return (cos(A + deltaX) - cos(A - deltaX)) / (2 * deltaX);
}

int PrimeCount(int A, int B){
    int prime_count = 0;
    //int * arr = (int *) malloc(B * sizeof(int));
    int arr[10000];
    for (int i = 0; i <= B; i++){
        arr[i] = i;
    }
    arr[1] = 0;
```

```

    for (int i = 2; i <= B; i++){
        if (arr[i] != 0){
            for (int j = i*i; j <= B; j += i){
                arr[j] = 0;
            }
            if(i >= A)
                prime_count++;
        }
    }
    //free(arr);
    return prime_count;
}

```

## main1.c

```

#include "lib.h"
#include <stdio.h>

void call_Derivative(){
    float A, deltaX;
    scanf("%f %f", &A, &deltaX);
    printf("%f\n", Derivative(A, deltaX));
}

void call_PrimeCount(){
    int A, B;
    scanf("%d %d", &A, &B);
    printf("%d\n", PrimeCount(A, B));
}

void call_error(){
    printf("exception during command processing\n");
}

int main(){
    int command;
    while(scanf("%d", &command) != EOF){
        switch(command){
            case 1:
                call_Derivative();
                fflush(stdin);
                break;
            case 2:
                call_PrimeCount();
                fflush(stdin);
                break;
            default:
                call_error();
                fflush(stdin);
        }
    }
}

```

```
    return 0;
}
```

## main2.c

```
#include <stdio.h>
#include <dlfcn.h>
#include <stdlib.h>
#include "lib.h"

typedef float (*Derivative_function)(float A , float deltaX);
typedef int (*PrimeCount_function)(int A, int B);
const char lib1_path[] = "./libmylib1x.so";
const char lib2_path[] = "./libmylib2x.so";

void change_handle(void ** lib_handle, int current_lib){
    if (current_lib == 0){
        *lib_handle = dlopen(lib1_path, RTLD_NOW);
    } else {
        *lib_handle = dlopen(lib2_path, RTLD_NOW);
    }
}

void change_function(Derivative_function *Derivative, PrimeCount_function *PrimeCount, void* lib_handle){
    *Derivative = dlsym(lib_handle, "Derivative");
    *PrimeCount = dlsym(lib_handle, "PrimeCount");
}

void call_error(){
    printf("exception during command processing\n");
}

int main(){
    int command, current_lib = 0;
    void * lib_handle;
    Derivative_function Derivative;
    PrimeCount_function PrimeCount;
    change_handle(&lib_handle, current_lib);
    change_function(&Derivative, &PrimeCount, lib_handle);
    while(scanf("%d", &command) != EOF){
        switch(command){
            case 0:
                printf("Called library changing\n");
                dlclose(lib_handle);
                current_lib = (current_lib + 1) % 2;
                change_handle(&lib_handle, current_lib);
                change_function(&Derivative, &PrimeCount, lib_handle);
                break;
            case 1:
                printf("Called Derivative function\n");
                float fA , deltaX;
```

```
        scanf("%f %f", &fA, &deltaX);
        printf("Result: %f\n", Derivative(fA, deltaX));
        break;
    case 2:
        printf("Called PrimeCount function\n");
        int A, B;
        scanf("%d %d", &A, &B);
        printf("Result: %d\n", PrimeCount(A, B));
        break;
    default:
        printf("%d\n", command);
        call_error();
    }
}
return 0;
}
```

## 10. Протокол strace

```
root@Anton-Sinyukov:/mnt/c/Users/sinyu/Desktop/os_5# strace ./main2
execve("./main2", ["/.main2"], 0x7ffce6dd05c0 /* 20 vars */) = 0
brk(NULL)                = 0x55fbdd13c000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffce8ece8a0) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1cd1fd9000
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=16355, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 16355, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1cd1fd5000
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\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\0\0\0\0\0\0\0..." , 784, 64) = 784
pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNUM\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0\0\0\0\0..." , 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340..." , 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2216304, ...}, 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\0\0\0\0\0\0\0..." , 784, 64) = 784
mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f1cd1dad000
mmap(0x7f1cd1dd5000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f1cd1dd5000
mmap(0x7f1cd1df6a000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xbdb000) = 0x7f1cd1df6a000
mmap(0x7f1cd1ffc2000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f1cd1ffc2000
mmap(0x7f1cd1fc8000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f1cd1fc8000
close(3)                                           = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1cd1daa000
arch_prctl(ARCH_SET_FS, 0x7f1cd1daa740) = 0
set_tid_address(0x7f1cd1daaa10)                 = 701
set_robust_list(0x7f1cd1daaa20, 24)             = 0
rseq(0x7f1cd1dab0e0, 0x20, 0, 0x53053053)      = 0
mprotect(0x7f1cd1fc2000, 16384, PROT_READ)       = 0
mprotect(0x55fbdcf45000, 4096, PROT_READ)        = 0
mprotect(0x7f1cd2013000, 8192, PROT_READ)        = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f1cd1fd5000, 16355)                   = 0
getrandom("\x08\x60\x20\x63\xdf\x84\xaa\x99", 8, GRND_NONBLOCK) = 8
brk(NULL)                                         = 0x55fbdd13c000
brk(0x55fbdd15d000)                             = 0x55fbdd15d000
openat(AT_FDCWD, "./libmylib1x.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0..." , 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15520, ...}, AT_EMPTY_PATH) = 0
getcwd("/mnt/c/Users/sinyu/Desktop/os_5", 128)  = 32
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f1cd1da5000
mmap(0x7f1cd1da6000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7f1cd1da6000
mmap(0x7f1cd1da7000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f1cd1da7000
mmap(0x7f1cd1da8000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f1cd1da8000
close(3)                                           = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=16355, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 16355, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1cd1fd5000
close(3)                                           = 0
```



```

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\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) = 0x7f1cd1cbe000
mmap(0x7f1cd1ccc000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f1cd1ccc000
mmap(0x7f1cd1d48000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f1cd1d48000
mmap(0x7f1cd1da3000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7f1cd1da3000
close(3) = 0
mprotect(0x7f1cd1da3000, 4096, PROT_READ) = 0
mprotect(0x7f1cd1da8000, 4096, PROT_READ) = 0
munmap(0x7f1cd1fd5000, 16355) = 0
newfstatat(0, "", {st_mode=S_IFCHR|0600, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH) = 0
read(0, 1 50 0.0001
"1 50 0.0001\n", 1024) = 12
newfstatat(1, "", {st_mode=S_IFCHR|0600, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH) = 0
write(1, "Called Derivative function\n", 27Called Derivative function
) = 27
write(1, "0.260182\n", 90.260182
) = 9
read(0, 2 1 100
"2 1 100\n", 1024) = 8
write(1, "Called PrimeCount function\n", 27Called PrimeCount function
) = 27
write(1, "25\n", 325
) = 3
read(0, 0
"0\n", 1024) = 2
write(1, "Called library changing\n", 24Called library changing
) = 24
munmap(0x7f1cd1da5000, 16432) = 0
munmap(0x7f1cd1cbe000, 942344) = 0
openat(AT_FDCWD, "./libmylib2x.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\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=15592, ...}, AT_EMPTY_PATH) = 0
getcwd("/mnt/c/Users/sinyu/Desktop/os_5", 128) = 32
mmap(NULL, 16440, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f1cd1da5000
mmap(0x7f1cd1da6000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7f1cd1da6000
mmap(0x7f1cd1da7000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f1cd1da7000
mmap(0x7f1cd1da8000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f1cd1da8000
close(3) = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=16355, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 16355, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1cd1fd5000
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\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) = 0x7f1cd1cbe000
mmap(0x7f1cd1ccc000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f1cd1ccc000
mmap(0x7f1cd1d48000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f1cd1d48000
mmap(0x7f1cd1da3000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7f1cd1da3000
close(3) = 0
mprotect(0x7f1cd1da3000, 4096, PROT_READ) = 0
mprotect(0x7f1cd1da8000, 4096, PROT_READ) = 0
munmap(0x7f1cd1fd5000, 16355) = 0
read(0, 1 50 0.01
"1 50 0.01\n", 1024) = 10
write(1, "Called Derivative function\n", 27Called Derivative function
) = 27
write(1, "0.262326\n", 90.262326
) = 9
read(0, "", 1024) = 0
exit_group(0) = ?
+++ exited with 0 +++
root@Anton-Sinyukov:/mnt/c/Users/sinyu/Desktop/os_5#

```