



Факультет информационных технологий и прикладной математики  
Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу  
«Операционные системы»**

Группа: М80 – 201Б-19  
Студент: Цыкин И.А.  
Преподаватель: Миронов Е.С.  
Оценка: \_\_\_\_\_  
Дата: \_\_\_\_\_

## **Содержание**

- 1 Постановка задачи
- 2 Общие сведения о программе
- 3 Общий метод и алгоритм решения
- 4 Листинг программы
- 5 Результаты работы программы
- 6 Strace
- 7 Вывод

## 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 использует 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 обеих 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.

№	Описание	Сигнатура	Реализация 1 Р	Реализация 2
3	Подсчёт количества простых чисел на отрезке [A, B] (A, B - натуральные)	Int PrimeCount(int A, int B)	Наивный алгоритм. Проверить делимость текущего числа на все предыдущие числа.	Решето Эратосфена
8	Перевод числа x из десятичной системы счисления в другую	Char* translation(long x)	Другая система счисления двоичная	Другая система счисления троичная

### Общие сведения о программе

Код состоит из 2 файлов библиотек и 2 файлов программ. Для решения данной лабораторной работы была использована библиотека `dlfcn.h`, она служит Для реализации динамической загрузки существует интерфейс динамической загрузки (Dynamic Loading API), дающий приложению пользователя возможность использовать совместно используемые библиотеки.

Основные функции:

1. **`void *dlopen(const char *filename, int flag);`** - загружает динамическую библиотеку, имя которой указано в строке *filename*, и возвращает прямой указатель на начало динамической библиотеки. Если *filename* не является полным именем файла
2. **`void *dlsym(void *handle, char *symbol);`** - функция, возвращения адреса, по которому символ расположен в памяти
3. **`const char *dlerror(void);`** - функция описания ошибок в случае неудачи
4. **`int dlclose(void *handle);`** - выгружает динамическую библиотеку

## Общий метод и алгоритм решения.

В динамических библиотеках описаны функция, которые подсчитывают количество простых чисел в промежутке и перевод числа в другие системы счисления. С помощью программ main1.c и main2.c можно осуществить работу с этими функциями. Разница в работе двух программ заключается в том, что main1.c загружает библиотеку, используя знания полученные на этапе компиляции; main2.c загружает библиотеки, используя только их местоположение и контракты;

## Листинг программы

### library1.c

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

bool prime(int n){
    if(n == 0 || n == 1){
        return false;
    }
    int b = sqrt(n);
    for(int i = 2; i <= b; ++i){
        if(n%i == 0){
            return false;
        }
    }
    return true;
}

int PrimeCount(int a, int b){
    int n = 0;
    printf("(Usual way)");
    for(int i = a; i < b + 1; ++i){
        if(prime(i)){
            n++;
        }
    }
    return n;
}

char* translation(long x){
    printf("(Binary)");
    char* res = (char*)malloc(64 * sizeof(char));
    do{
        *--res = x % 2 + '0';
        x /= 2;
    }while (x != 0);
    return res;
}
```

```

}

int main(){
    return 0;
}

```

## library2.c

```

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int PrimeCount(int a, int b){
    int n = 0;
    printf("(The sieve of Eratosthenes)");
    int mas[b+1];
    for(int i = 0; i < b+1; ++i){
        mas[i] = i;
    }
    for(int i = 2; i*i <= b; ++i){
        if(mas[i]){
            for(int j = i*i; j <= b; j += i){
                mas[j] = 0;
            }
        }
    }
    for(int i = 0; i <= b; ++i){
        if(mas[i] && i >= a && i != 1){
            n++;
        }
    }
    return n;
}

char* translation(long x){
    printf("(Trenary)");
    char* res = (char*)malloc(64 * sizeof(char));
    do{
        *--res = x % 3 + '0';
        x /= 3;
    }while (x != 0);
    return res;
}

int main(){
    return 0;
}

```

## main1.c

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

int main(int argc, char* argv[]) {
    int x, a, b;
    long c;
    for(;;){
        scanf("%d", &x);
        if(x == 1){
            scanf("%d %d", &a, &b);
            printf("Result ");
            int n = PrimeCount(a, b);
            printf("): %d\n", n);
        }else if(x == 2){
            scanf("%ld", &c);
            printf("Result ");
            char* res = translation(c);
            printf("): %s\n", res);
        }else{
            return 0;
        }
    }
}
```

## main2.c

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

#define LIBRARY1 "./library1.so"
#define LIBRARY2 "./library2.so"

int main(int argc, char* argv[]) {
    void *library;
    int x, a, b;
    long c;

    if(argv[1]){
        x = atoi(argv[1]);
    }

    if(x == 0){
        library = dlopen(LIBRARY2, RTLD_LAZY);
    } else{
        library = dlopen(LIBRARY1, RTLD_LAZY);
    }
}
```

```

if (!library) {
    printf("Error dlopen(): %s\n", dlerror());
    return 1;
}

int(*PrimeCount)(int x, int y) = dlsym(library, "PrimeCount");
char*(translation)(long x) = dlsym(library, "translation");

for(;;){
    scanf("%d", &x);
    if(x == 1){
        scanf("%d %d", &a, &b);
        printf("Result ");
        int n = PrimeCount(a, b);
        printf("): %d\n", n);
    }else if(x == 2){
        scanf("%ld", &c);
        printf("Result ");
        char* res = translation(c);
        printf("): %s\n", res);
    }else{
        dlclose(library);
        return 0;
    }
}
}

```

## Результаты работы программы

```

vaney@V-box:$ cd */os_lab5
vaney@V-box:~/Examples/os_lab5$ ./main1
vaney@V-box:~/Examples/os_lab5$ gcc -o main1 main1.c -lm
vaney@V-box:~/Examples/os_lab5$ ./main1
1 2 17
Result (Usual way): 7
1 44 88
Result (Usual way): 9
2 22
Result (Binary): 10110
2 4
Result (Binary): 100
4
vaney@V-box:~/Examples/os_lab5$ gcc -fPIC -c library1.c -lm -o library1.o
vaney@V-box:~/Examples/os_lab5$ gcc -fPIC -c library2.c -lm -o library2.o
vaney@V-box:~/Examples/os_lab5$ gcc -shared -o library1.so library1.o -lm
vaney@V-box:~/Examples/os_lab5$ gcc -shared -o library2.so library2.o -lm
vaney@V-box:~/Examples/os_lab5$ gcc -o main2 main2.c -lm -ldl
vaney@V-box:~/Examples/os_lab5$ ./main2
1 2 17
Result (Usual way): 7
1 44 88
Result (Usual way): 9
2 22
Result (Binary): 10110
2 4
Result (Binary): 100
4

```



```

vaney@V-box:~/Examples/os_lab5$ ./main2 0
1 2 17
Result (The sieve of Eratosthenes): 7
1 44 88
Result (The sieve of Eratosthenes): 9
2 22
Result (Trenary): 211
2 4
Result (Trenary): 11
4
vaney@V-box:~/Examples/os_lab5$

```

## Strace

```

vaney@V-box:~/Examples/os_lab5$ strace ./main1
execve("./main1", [ "./main1" ], 0x7ffe7b82f820 /* 58 vars */) = 0
brk(NULL)                               = 0x561f86a6b000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffed64a84c0) = -1 EINVAL (Invalid
argument)
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
fstat(3, {st_mode=S_IFREG|0644, st_size=67999, ...}) = 0
mmap(NULL, 67999, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f211255b000
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\0\0\0\3\0>\
0\1\0\0\0\300\363\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1369352, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f2112559000
mmap(NULL, 1368336, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f211240a000
mmap(0x7f2112419000, 684032, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x0) = 0x7f2112419000
mmap(0x7f21124c0000, 618496, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xb6000) = 0x7f21124c0000
mmap(0x7f2112557000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x14c000) = 0x7f2112557000
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\0\0\0\3\0>\0\1\0\0\0\360q\2\0\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"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\
0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\
355Y\377\t\334"... , 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 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"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\
0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\
355Y\377\t\334"... , 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f2112218000

```

```

mprotect(0x7f211223d000, 1847296, PROT_NONE) = 0
mmap(0x7f211223d000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x25000) = 0x7f211223d000
mmap(0x7f21123b5000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x19d000) = 0x7f21123b5000
mmap(0x7f2112400000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1e7000) = 0x7f2112400000
mmap(0x7f2112406000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7f2112406000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f2112215000
arch_prctl(ARCH_SET_FS, 0x7f2112215740) = 0
mprotect(0x7f2112400000, 12288, PROT_READ) = 0
mprotect(0x7f2112557000, 4096, PROT_READ) = 0
mprotect(0x561f85f2e000, 4096, PROT_READ) = 0
mprotect(0x7f2112599000, 4096, PROT_READ) = 0
munmap(0x7f211255b000, 67999) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
brk(NULL) = 0x561f86a6b000
brk(0x561f86a8c000) = 0x561f86a8c000
read(0, 1 2 17
"1 2 17\n", 1024) = 7
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "Result (Usual way): 7\n", 22Result (Usual way): 7
) = 22
read(0, 2 22
"2 22\n", 1024) = 5
write(1, "Result (Binary): 10110\n", 23Result (Binary): 10110
) = 23
read(0, 4
"4\n", 1024) = 2
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++
vaney@V-box:~/Examples/os_lab5$ strace ./main2 0
execve("./main2", [ "./main2", "0"], 0x7ffc54ee8368 /* 58 vars */) = 0
brk(NULL) = 0x55e12fc3a000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffecd7c57e0) = -1 EINVAL (Invalid
argument)
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
fstat(3, {st_mode=S_IFREG|0644, st_size=67999, ...}) = 0
mmap(NULL, 67999, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fea0884d000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 \22\0\0\0\0\0\0",
832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=18816, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fea0884b000
mmap(NULL, 20752, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fea08845000

```

```
mmmap(0x7fea08846000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1000) = 0x7fea08846000
mmap(0x7fea08848000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x3000) = 0x7fea08848000
mmap(0x7fea08849000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x3000) = 0x7fea08849000
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\0\3\0>\0\1\0\0\0\360q\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"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\\
0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\\
355Y\377\t\334"... , 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 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"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\\
0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\\
355Y\377\t\334"... , 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fea08653000
mprotect(0x7fea08678000, 1847296, PROT_NONE) = 0
mmap(0x7fea08678000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x25000) = 0x7fea08678000
mmap(0x7fea087f0000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x19d000) = 0x7fea087f0000
mmap(0x7fea0883b000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1e7000) = 0x7fea0883b000
mmap(0x7fea08841000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7fea08841000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fea08650000
arch_prctl(ARCH_SET_FS, 0x7fea08650740) = 0
mprotect(0x7fea0883b000, 12288, PROT_READ) = 0
mprotect(0x7fea08849000, 4096, PROT_READ) = 0
mprotect(0x55e12f41c000, 4096, PROT_READ) = 0
mprotect(0x7fea0888b000, 4096, PROT_READ) = 0
munmap(0x7fea0884d000, 67999) = 0
brk(NULL) = 0x55e12fc3a000
brk(0x55e12fc5b000) = 0x55e12fc5b000
openat(AT_FDCWD, "./library2.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\240\20\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=16384, ...}) = 0
getcwd("/home/vaney/Examples/os_lab5", 128) = 29
mmap(NULL, 16448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fea08859000
mmap(0x7fea0885a000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1000) = 0x7fea0885a000
mmap(0x7fea0885b000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x2000) = 0x7fea0885b000
```

```

mmap(0x7fea0885c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7fea0885c000
close(3) = 0
mprotect(0x7fea0885c000, 4096, PROT_READ) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
read(0, 1 2 17
"1 2 17\n", 1024) = 7
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "Result (The sieve of Eratosthene"... , 38Result (The sieve of
Eratosthenes): 7
) = 38
read(0, 2 22
"2 22\n", 1024) = 5
write(1, "Result (Trenary): 211\n", 22Result (Trenary): 211
) = 22
read(0, 4
"4\n", 1024) = 2
munmap(0x7fea08859000, 16448) = 0
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++
vaney@V-box:~/Examples/os_lab5$

```

## Вывод

Выполняя данную лабораторную работу, я узнал о том, что такое динамические библиотеки, а так же научился использовать их. По моему мнению, и статическое, и динамическое подключение библиотек равносильно, так у каждой из них есть преимущества. К примеру: динамические библиотеки помогают экономить достаточное количество памяти на жестком диске и в оперативной памяти, нет необходимости подключать одну большую библиотеку, тем самым вытащив из нее пару функций или методов. Но с другой стороны, статические библиотеки более простые и работа с функциями и методами не доставляет больших хлопот.