Московский авиационный институт (национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование» Дисциплина «Операционные системы»

Лабораторная работа №5

Тема: Динамические библиотеки

Студент: Будникова В.П.

Группа: М8О-207Б-19

Преподаватель: Миронов Е. С.

Дата:

Оценка:

Москва, 2020

Цель работы: приобретение практических навыков в создании динамических библиотек и создании программ, которые используют функции динамических библиотек

Задача: 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 итоге, в лабораторной работе необходимо получить следующие части:

1. Динамические библиотеки, реализующие контракты, которые заданы вариантом;

2. Тестовая программа (программа No1), которая используют одну из библиотек, используя знания полученные на этапе компиляции;

3. Тестовая программа (программа No2), 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 для программы No2).

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(20):

Описание	Сигнатура	Реализация1	Реализация2
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
Перевод числа x из десятичной системы счисления в другую	Char* translation(long x)	Другая система счисления двоичная	Другая система счисления троичная

Листинг программы

h.h

```
#ifndef H_H
#define H_H
#include <stddef.h>
extern int CountPrimeNum(int a, int b);
extern char * Trans(long a);
#endif
```

implement1.c

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

int CountPrimeNum(int a, int b) {
    int rez = 0;
    int fl;
    for (int i = a; i < b + 1; ++i) {
        fl = 0;
        for (int j = 2; j < i; ++j) {
            if (i % j == 0) {
                fl = 1;
            }
        }
        if (fl == 0) {
```

```
            ++rez;
        }
    }
    if (a == 1 && rez != 0) {
        --rez;
    }
    return rez;
}

char * Trans(long a) {
    char * rez;
    rez = malloc(sizeof(long) * 8);
    int f = 0;
    if (a < 0) {
        a *= -1;
        f = 1;
    }
    for (int i = sizeof(long) * 8 - 1; i >= 0; --i) {
        if ((a & 1) == 1) {
            rez[i] = '1';
        } else {
            rez[i] = '0';
        }
        a >>= 1;
    }
    if (f) {
        rez[0] = '-';
    }
    return rez;
}
```

implement2.c

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

int CountPrimeNum(int a, int b) {
    int rez = 0;
    int * mas = (int *)malloc(sizeof(int) * (b + 2));
    for (int i = 0; i < b + 1; ++i) {
        mas[i] = i;
    }
    mas[1] = 0;
    for (int i = 2; i < b + 1; i++) {
        if (mas[i] != 0) {
            if (i >= a && i <= b) {
                rez++;
            }
            for (int j = i * i; j < b + 1; j += i) {
                mas[j] = 0;
            }
        }
```

```
    }
    free(mas);
    return rez;
}

char * Trans(long a) {
    char * rez;
    rez = malloc(sizeof(long) * 6);
    int f = 0;
    if (a < 0) {
        a *= -1;
        f = 1;
    }
    for (int i = sizeof(long) * 6 - 1; i >= 0; --i) {
        if (a % 3 == 0) {
            rez[i] = '0';
        } else if (a % 3 == 1) {
            rez[i] = '1';
        } else {
            rez[i] = '2';
        }
        a /= 3;
    }
    if (f) {
        rez[0] = '-';
    }
    return rez;
}
```

main1.c

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

int main() {
    int x;
    while (scanf("%d", &x) > 0) {
        if (x == 1) {
            int a;
            int b;
            scanf("%d%d", &a, &b);
            printf("%d\n", CountPrimeNum(a, b));
        } else if (x == 2) {
            long a;
            scanf("%ld", &a);
            char * str = Trans(a);
            printf("%s\n", str);
            free(str);
        }
    }
}
```

main2.c

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

int main() {
    char * lib1 = "/Users/valeriabudnikova/Desktop/ALL/3S/OC/lab5/libim1.so";
    char * lib2 = "/Users/valeriabudnikova/Desktop/ALL/3S/OC/lab5/libim2.so";
    char * nameCount = "CountPrimeNum";
    char * nameTr = "Trans";

    void * handle;

    int (*CountPrimeNum) (int, int);
    char * (*Trans) (long);

    handle = dlopen(lib1, RTLD_LAZY);
    if (!handle) {
        printf("error_open");
        exit(EXIT_FAILURE);
    }

    char * error;
    CountPrimeNum = dlsym(handle, nameCount);
    Trans = dlsym(handle, nameTr);

    if ((error = dlerror()) != NULL)  {
        printf("error_dlsym");
        exit(EXIT_FAILURE);
    }

    int x;
    int f = 1;
    while (scanf("%d", &x) > 0) {
        if (x == 0) {
            int err = dlclose(handle);
            if (err != 0) {
                printf("error_close");
                exit(EXIT_FAILURE);
            }
            if (f == 1) {
                handle = dlopen(lib2, RTLD_LAZY);
                f = 2;
            } else {
                handle = dlopen(lib1, RTLD_LAZY);
                f = 1;
            }
            if (!handle) {
                printf("error_open");
                exit(EXIT_FAILURE);
```

```
            }
            CountPrimeNum = dlsym(handle, nameCount);
            Trans = dlsym(handle, nameTr);

            if ((error = dlerror()) != NULL)  {
               printf("error_dlsym");
               exit(EXIT_FAILURE);
            }
        } else if (x == 1) {
            int a;
            int b;
            scanf("%d%d", &a, &b);
            printf("%d\n", (*CountPrimeNum)(a, b));
        } else if (x == 2) {
            long a;
            scanf("%ld", &a);
            printf("%s\n", (*Trans)(a));
        }
    }
    int err = dlclose(handle);
    if (err != 0) {
        printf("error_close");
        exit(EXIT_FAILURE);
    }
}
```

makefile

```
rez: implement1 implement2 main1
	gcc -std=c11 main1.o implement1.o -o rez1
	gcc -std=c11 main1.o implement2.o -o rez2

main1: main1.c
	gcc -std=c11 -c main1.c

rez3:	main2 implement1 implement2
	gcc -std=c11 main2.o -o rez3 -ldl

main2: main2.c
	gcc -std=c11 -c main2.c

implement1: h.h impl1
	gcc -std=c11 -shared -o libim1.so implement1.o

implement2: h.h impl2
	gcc -std=c11 -shared -o libim2.so implement2.o

impl1:
	gcc -std=c11 -fPIC -c implement1.c

impl2:
	gcc -std=c11 -fPIC -c implement2.c

clean:
	rm -rf *.o *.so rez1 rez2 rez3
```

Makefile:

При компиляции используется ключ -fPIC(Position Independent Code), чтобы функции в библиотеке использовали относительную адресацию (так как при обычной компиляции созданный объектный файл "не знает" в каких адресах будет лежать использующая их программа). Ключ -shared используется для создания динамической библиотеки.

Тесты и протокол исполнения

```
Lera:lab5 valeriabudnikova$ make rez
gcc -std=c11 -fPIC -c implement1.c
gcc -std=c11 -shared -o libim1.so implement1.o
gcc -std=c11 -fPIC -c implement2.c
gcc -std=c11 -shared -o libim2.so implement2.o
gcc -std=c11 -c main1.c
gcc -std=c11 main1.o implement1.o -o rez1
gcc -std=c11 main1.o implement2.o -o rez2
Lera:lab5 valeriabudnikova$ ./rez1
1 2 100
25
1 2 2
1
1 50 1000
153
2 12
00000000000000000000000000000000000000000000000000000000001100
2 2
00000000000000000000000000000000000000000000000000000000000010
2 32
00000000000000000000000000000000000000000000000000000100000
^C
Lera:lab5 valeriabudnikova$ ./rez2
1 2 100
25
1 2 2
1
1 50 1000
153
2 12
00000000000000000000000000000000000000000000000110
2 2
00000000000000000000000000000000000000000000000002
2 32
00000000000000000000000000000000000000000000001012
2 3
00000000000000000000000000000000000000000000000010
2 9
00000000000000000000000000000000000000000000000100
^C
Lera:lab5 valeriabudnikova$ make rez3
gcc -std=c11 -c main2.c
gcc -std=c11 -fPIC -c implement1.c
gcc -std=c11 -shared -o libim1.so implement1.o
gcc -std=c11 -fPIC -c implement2.c
gcc -std=c11 -shared -o libim2.so implement2.o
gcc -std=c11 main2.o -o rez3 -ldl
```

```
Lera:lab5 valeriabudnikova$ ./rez3
1 2 100
25
0
1 2 100
25
0
2 100
00000000000000000000000000000000000000000000000000000001100100
0
2 100
00000000000000000000000000000000000000000000010201
2 30
00000000000000000000000000000000000000000000001010
0
2 30
00000000000000000000000000000000000000000000000000000011110
^C
```

Вывод strace

```
strace ./rez1 < 1.txt
execve("./rez1", ["./rez1"], 0x7fffe16851c0 /* 67 vars */) = 0
brk(NULL)                               = 0x563a138e8000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe12918780) = -1 EINVAL (Недопустимый аргумент)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=118993, ...}) = 0
mmap(NULL, 118993, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fc051b02000
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\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
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fc051b00000
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) = 0x7fc05190e000
mprotect(0x7fc051933000, 1847296, PROT_NONE) = 0
mmap(0x7fc051933000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x25000) = 0x7fc051933000
mmap(0x7fc051aab000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) =
0x7fc051aab000
mmap(0x7fc051af6000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1e7000) = 0x7fc051af6000
mmap(0x7fc051afc000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7fc051afc000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7fc051b01540) = 0
mprotect(0x7fc051af6000, 12288, PROT_READ) = 0
mprotect(0x563a11e12000, 4096, PROT_READ) = 0
mprotect(0x7fc051b4d000, 4096, PROT_READ) = 0
munmap(0x7fc051b02000, 118993)          = 0
fstat(0, {st_mode=S_IFREG|0664, st_size=35, ...}) = 0
brk(NULL)                               = 0x563a138e8000
brk(0x563a13909000)                     = 0x563a13909000
read(0, "1 2 100\n2 12  \n1 3 30\n1 35 100\n2"..., 4096) = 35
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "25\n", 325
```

```
)                   = 3
write(1, "00000000000000000000000000000000"...,
650000000000000000000000000000000000000000000000000000000000000001100
) = 65
write(1, "9\n", 29
)                   = 2
write(1, "14\n", 314
)                   = 3
read(0, "", 4096)                       = 0
write(1, "00000000000000000000000000000000"...,
650000000000000000000000000000000000000000000000000000000000000100011
) = 65
exit_group(0)                           = ?
+++ exited with 0 +++
strace ./rez2 < 1.txt
execve("./rez2", ["./rez2"], 0x7ffe8b0885d0 /* 67 vars */) = 0
brk(NULL)                               = 0x55678d996000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff02287b00) = -1 EINVAL (Недопустимый аргумент)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=118993, ...}) = 0
mmap(NULL, 118993, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f15c4efa000
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"..., 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", 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
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f15c4ef8000
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"..., 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", 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) = 0x7f15c4d06000
mprotect(0x7f15c4d2b000, 1847296, PROT_NONE) = 0
mmap(0x7f15c4d2b000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x25000) = 0x7f15c4d2b000
mmap(0x7f15c4ea3000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) =
0x7f15c4ea3000
mmap(0x7f15c4eee000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1e7000) = 0x7f15c4eee000
mmap(0x7f15c4ef4000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f15c4ef4000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f15c4ef9540) = 0
mprotect(0x7f15c4eee000, 12288, PROT_READ) = 0
mprotect(0x55678d619000, 4096, PROT_READ) = 0
mprotect(0x7f15c4f45000, 4096, PROT_READ) = 0
munmap(0x7f15c4efa000, 118993)          = 0
fstat(0, {st_mode=S_IFREG|0664, st_size=35, ...}) = 0
brk(NULL)                               = 0x55678d996000
brk(0x55678d9b7000)                     = 0x55678d9b7000
read(0, "1 2 100\n2 12  \n1 3 30\n1 35 100\n2"..., 4096) = 35
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "25\n", 325
)                   = 3
write(1, "00000000000000000000000000000000"..., 490000000000000000000000000000000000000000000000000000000000000000110
) = 49
write(1, "9\n", 29
)                   = 2
write(1, "14\n", 314
)                   = 3
read(0, "", 4096)                       = 0
write(1, "00000000000000000000000000000000"..., 490000000000000000000000000000000000000000000000000000000000000001022
) = 49
exit_group(0)                           = ?
```

```
+++ exited with 0 +++
strace ./rez3 < 2.txt
execve("./rez3", ["./rez3"], 0x7fff0b0c5710 /* 67 vars */) = 0
brk(NULL)                               = 0x56444d206000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc9e2a0be0) = -1 EINVAL (Недопустимый аргумент)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=118993, ...}) = 0
mmap(NULL, 118993, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f810e2fd000
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\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) = 0x7f810e2fb000
mmap(NULL, 20752, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f810e2f5000
mmap(0x7f810e2f6000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1000) = 0x7f810e2f6000
mmap(0x7f810e2f8000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) =
0x7f810e2f8000
mmap(0x7f810e2f9000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x3000) = 0x7f810e2f9000
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\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) = 0x7f810e103000
mprotect(0x7f810e128000, 1847296, PROT_NONE) = 0
mmap(0x7f810e128000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x25000) = 0x7f810e128000
mmap(0x7f810e2a0000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) =
0x7f810e2a0000
mmap(0x7f810e2eb000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1e7000) = 0x7f810e2eb000
mmap(0x7f810e2f1000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f810e2f1000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f810e100000
arch_prctl(ARCH_SET_FS, 0x7f810e100740) = 0
mprotect(0x7f810e2eb000, 12288, PROT_READ) = 0
mprotect(0x7f810e2f9000, 4096, PROT_READ) = 0
mprotect(0x56444bc3f000, 4096, PROT_READ) = 0
mprotect(0x7f810e348000, 4096, PROT_READ) = 0
munmap(0x7f810e2fd000, 118993)          = 0
brk(NULL)                               = 0x56444d206000
brk(0x56444d227000)                     = 0x56444d227000
openat(AT_FDCWD, "/home/lera/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/lera/5/libim1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`\20\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=16144, ...}) = 0
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f810e316000
mmap(0x7f810e317000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1000) = 0x7f810e317000
mmap(0x7f810e318000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f810e318000
mmap(0x7f810e319000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x2000) = 0x7f810e319000
close(3)                                = 0
mprotect(0x7f810e319000, 4096, PROT_READ) = 0
fstat(0, {st_mode=S_IFREG|0664, st_size=37, ...}) = 0
```

```
read(0, "1 2 100\n2 12\n0  \n1 3 30\n1 35 100"..., 4096) = 37
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "25\n", 325
)                       = 3
write(1, "00000000000000000000000000000000"...,
650000000000000000000000000000000000000000000000000000000000001100
) = 65
munmap(0x7f810e316000, 16432)           = 0
openat(AT_FDCWD, "/home/lera/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/lera/5/libim2.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200\20\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=16200, ...}) = 0
mmap(NULL, 16440, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f810e316000
mmap(0x7f810e317000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1000) = 0x7f810e317000
mmap(0x7f810e318000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f810e318000
mmap(0x7f810e319000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x2000) = 0x7f810e319000
close(3)                                = 0
mprotect(0x7f810e319000, 4096, PROT_READ) = 0
write(1, "9\n", 29
)                       = 2
write(1, "14\n", 314
)                       = 3
read(0, "", 4096)                       = 0
write(1, "00000000000000000000000000000000"..., 490000000000000000000000000000000000000000000001022
) = 49
munmap(0x7f810e316000, 16440)           = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Strace:

openat - системный вызов, который работает аналогично open, но при определенных аргументах меняется относительный путь к файлу(если такой был задан).

write - системный вызов, для записи в файл. Первый аргумент - файловый дескриптор. Второй аргумент - адрес, с которого начинается буфер. Третий аргумент - количество байтов, которое нужно записать в файл, на который ссылается файловый дескриптор из буфера. При удачном завершении возвращается количество байтов, которые были записаны(в случае ошибки - "-1").

Выводы

В данной лабораторной работе я научилась создавать динамические библиотеки и использовать их в программе. Самым сложным в данной лабораторной работе оказалась собрать программу. Для этого потребовалось искать очень много информации в интернете. Работа с динамическими библиотеками показалась мне очень удобной, так как в самой программе мы можем сами "переподключать" библиотеки.

Список литературы

1. Таненбаум Э., Бос Х. *Современные операционные системы. — 4-е изд.* — СПб.: Издательский дом «Питер», 2018. — С. 111 - 123

2. Поисковик Google [электронный ресурс] URL: https://google.com/ (дата обращения: 22.09.2020)