

**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: Глушатов И.С. 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 №1), которая использует одну из библиотек, используя знания полученные на этапе компиляции;
3. 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 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 3

Описание	Сигнатура	Реализация 1	Реализация 2
Расчёт значения числа $e$ (основание натурального логарифма)	Float E(int x)	$(1 + 1/x)^x$	Сумма ряда по $n$ от 0 до $x$ , где элементы ряда равны: $(1/(n!))$
Подсчёт наибольшего общего делителя для двух натуральных чисел	Int GCF(int A, int B)	Алгоритм Евклида	Наивный алгоритм. Пытаться разделить числа на все числа, что меньше $A$ и $B$ .

### Листинг программы

#### static\_link\_main.c

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

int main() {

    short command = 0;
    printf("1 - E\n2 - GCF\n\n: ");
    while(scanf("%hd", &command)>0) {

        if (command == 1) {
            int x;
            printf("Введите число: ");
            scanf("%d", &x);
            printf("Ответ: %f\n\n", E(x));
        } else if (command == 2) {
            int A, B;
            printf("Введите два натуральных числа: ");
            scanf("%d%d", &A, &B);
            printf("Ответ: %d\n\n", GCF(A, B));
        } else {
            printf("Неправильно введенная команда\n");
        }
    }
}
```

```

    }
    printf(": ");
}

return 0;
}

```

### **dynamic\_link\_main.c**

```

#include <dlfcn.h>
#include "e_and_gcf.h"

#define error_msg(msg) \
    do { perror(msg); exit(EXIT_FAILURE); } while(0)

int main() {

    short flag = 0;

    void* current_library = dlopen("./librealization1.so", RTLD_LAZY);
    if (!current_library) error_msg("Динамическая библиотека не смогла открыться");

    float (*E) (int);
    int (*GCF) (int, int);

    *(void**) (&E) = dlsym(current_library, "E");
    *(void**) (&GCF) = dlsym(current_library, "GCF");
    if (dlerror()) error_msg("Ошибка загрузки функций");

    short command = 0;
    printf("0 - Change library\n1 - E\n2 - GCF\n\n: ");
    while(scanf("%hd", &command)>0) {

        if (command == 0) {
            dlclose(current_library);
            if (flag) {
                current_library = dlopen("./librealization1.so", RTLD_LAZY);
                if (!current_library)
                    error_msg("Динамическая библиотека не смогла открыться");
                flag = 0;
            } else {
                current_library = dlopen("./librealization2.so", RTLD_LAZY);
                if (!current_library)
                    error_msg("Динамическая библиотека не смогла открыться");
                flag = 1;
            }
        }
    }
}

```

```

        *(void**) (&E) = dlsym(current_library, "E");
        *(void**) (&GCF) = dlsym(current_library, "GCF");
        if (dlerror()) error_msg("Ошибка загрузки функций");
        printf("Загрузка прошла успешно\n\n");
    } else if (command == 1) {
        int x;
        printf("Введите число: ");
        scanf("%d", &x);
        printf("Ответ: %f\n\n", E(x));
    } else if (command == 2) {
        int A, B;
        printf("Введите два натуральных числа: ");
        scanf("%d%d", &A, &B);
        printf("Ответ: %d\n\n", GCF(A, B));
    } else {
        printf("Неправильно введенная команда\n\n");
    }
    printf(": ");
}

dlclose(current_library);
return 0;
}

```

### **e\_and\_gcf.h**

```

#ifndef E_AND_GCF_H
#define E_AND_GCF_H

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

#define error_msg(msg) \
    do { perror(msg); exit(EXIT_FAILURE); } while(0)

float E(const int x);
int GCF(int A, int B);

#endif

```

## realization1.c

```
#include "e_and_gcf.h"
float E(const int x) {
    if (x < 1) error_msg("Число должно быть натуральным");
    printf("Подсчет E формулой (1+1/x)^x...\n");
    float res = 1;
    float k = (1 + 1/((float)x));
    for (int i = 0; i < x; ++i) res*=k;
    return res;
}

int GCF(int A, int B) {
    if (A < 1 || B < 1) error_msg("Аргументы должны быть натуральными числами!");
    printf("Подсчет НОД методом Евклида...\n");
    while (A != 0 && B != 0) {
        if (A > B) {
            A %= B;
        } else {
            B %= A;
        }
    }
    return A+B;
}
```

## realization2.c

```
#include "e_and_gcf.h"
float E(const int x) {
    if (x < 1) error_msg("Число должно быть натуральным");
    printf("Подсчет E суммой ряда...\n");
    float res = 1;
    float k = 1;
    for (int i = 1; i <= x; ++i) {
        k*=i;
        res+=1/k;
    }
    return res;
}

int GCF(const int A, const int B) {
    if (A < 1 || B < 1) error_msg("Аргументы должны быть натуральными числами!");
    printf("Подсчет НОД наивным методом\n");
    int result = A < B ? A : B;
    while (A % result != 0 || B % result != 0) {
        --result;
    }
    return result;
}
```

## Makefile

CC = gcc

FLAGS = -Wall -Werror -Wextra -pedantic -g

all: static\_link\_main1.out static\_link\_main2.out dynamic\_link\_main.out

static\_link\_main1.out: static\_link\_main.o realization1.o

    \${CC} \${FLAGS} static\_link\_main.o realization1.o -o static\_link\_main1.out

static\_link\_main2.out: static\_link\_main.o realization2.o

    \${CC} \${FLAGS} static\_link\_main.o realization2.o -o static\_link\_main2.out

static\_link\_main.o: static\_link\_main.c

    \${CC} \${FLAGS} -c static\_link\_main.c

realization1.o: realization1.c e\_and\_gcf.h

    \${CC} \${FLAGS} -c -fPIC realization1.c

realization2.o: realization2.c e\_and\_gcf.h

    \${CC} \${FLAGS} -c -fPIC realization2.c

dynamic\_link\_main.out: dynamic\_link\_main.o librealization1.so librealization2.so

    \${CC} \${FLAGS} -rdynamic -o dynamic\_link\_main.out dynamic\_link\_main.o -ldl -lm -  
librealization2 -lrealization1

dynamic\_link\_main.o: dynamic\_link\_main.c e\_and\_gcf.h

    \${CC} \${FLAGS} -c -fPIC dynamic\_link\_main.c

librealization1.so: realization1.o e\_and\_gcf.h

    \${CC} \${FLAGS} -shared -o librealization1.so realization1.o -lm

librealization2.so: realization2.o e\_and\_gcf.h

    \${CC} \${FLAGS} -shared -o librealization2.so realization2.o -lm

clean:

    rm -rf \*.so \*.o \*.out

## Тесты и протокол исполнения

igor@igor-Aspire-A315-53G:~/Рабочий стол/с/ОС/lab5\$ ./\*main1.out

1 - E

2 - GCF

: 1

Введите число: 34

Подсчет E формулой  $(1+1/x)^x$ ...

Ответ: 2.679357

: 1

Введите число: 3

Подсчет E формулой  $(1+1/x)^x$ ...

Ответ: 2.370371

: 2

Введите два натуральных числа: 45 225

Подсчет НОД методом Евклида...

Ответ: 45

:

igor@igor-Aspire-A315-53G:~/Рабочий стол/с/ОС/lab5\$ ./\*main2.out

1 - E

2 - GCF

: 1

Введите число: 45

Подсчет E суммой ряда...

Ответ: 2.718282

: 1

Введите число: 200

Подсчет E суммой ряда...

Ответ: 2.718282

: 1

Введите число: 3

Подсчет E суммой ряда...

Ответ: 2.666667

: 2

Введите два натуральных числа: 630 90

Подсчет НОД наивным методом

Ответ: 90

:



igor@igor-Aspire-A315-53G:~/Рабочий стол/с/ОС/lab5\$ ./\*main.out

0 - Change library

1 - E

2 - GCF

: 1

Введите число: 45

Подсчет E формулой  $(1+1/x)^x$ ...

Ответ: 2.688688

: 2

Введите два натуральных числа: 45 540

Подсчет НОД методом Евклида...

Ответ: 45

: 0

Загрузка прошла успешно

: 1

Введите число: 10

Подсчет E суммой ряда...

Ответ: 2.718282

: 2

Введите два натуральных числа: 55555 11

Подсчет НОД наивным методом

Ответ: 1

:

igor@igor-Aspire-A315-53G:~/Рабочий стол/с/ОС/lab5\$

### Вывод strace

igor@igor-Aspire-A315-53G:~/Рабочий стол/с/ОС/lab5\$ strace ./\*main.out

execve("./dynamic\_link\_main.out", ["/dynamic\_link\_main.out"], 0x7ffc3f273240 /\* 62 vars \*/) = 0

brk(NULL) = 0x55d6bfef2000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffef00e9ab0) = -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) = 0x7fc3949d3000

```

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\3\0>\0\1\0\0\0 \22\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)
= 0x7fc3949d1000
mmap(NULL, 20752, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fc3949cb000
mmap(0x7fc3949cc000, 8192, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fc3949cc000
mmap(0x7fc3949ce000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fc3949ce000
mmap(0x7fc3949cf000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fc3949cf000
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\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"..., 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
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"..., 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) =
0x7fc3947d9000
mprotect(0x7fc3947fe000, 1847296, PROT_NONE) = 0
mmap(0x7fc3947fe000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7fc3947fe000
mmap(0x7fc394976000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7fc394976000
mmap(0x7fc3949c1000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fc3949c1000
mmap(0x7fc3949c7000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fc3949c7000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7fc3947d6000
arch_prctl(ARCH_SET_FS, 0x7fc3947d6740) = 0
mprotect(0x7fc3949c1000, 12288, PROT_READ) = 0
mprotect(0x7fc3949cf000, 4096, PROT_READ) = 0
mprotect(0x55d6be080000, 4096, PROT_READ) = 0

```

```

mprotect(0x7fc394a1e000, 4096, PROT_READ) = 0
munmap(0x7fc3949d3000, 118993) = 0
brk(NULL) = 0x55d6bfef2000
brk(0x55d6bff13000) = 0x55d6bff13000
openat(AT_FDCWD, "./librealization1.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\240\20\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=19240, ...}) = 0
getcwd("/home/igor/\320\240\320\260\320\261\320\276\321\207\320\270\320\271\321\201\321\202\320\276\320\273/c/OC/lab5", 128) = 45
mmap(NULL, 16448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fc3949ec000
mmap(0x7fc3949ed000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fc3949ed000
mmap(0x7fc3949ee000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fc3949ee000
mmap(0x7fc3949ef000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fc3949ef000
close(3) = 0
mprotect(0x7fc3949ef000, 4096, PROT_READ) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "0 - Change library\n", 190 - Change library) = 19
write(1, "1 - E\n", 61 - E)
)

= 6
write(1, "2 - GCF\n", 82 - GCF)
) = 8
write(1, "\n", 1)
) = 1
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, ":", 2) = 2
read(0, 1
"1\n", 1024) = 2
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265\321\207\320\270\321\201\320\273\320\276:", 27Введите число: ) = 27
read(0, 56
"56\n", 1024) = 3

```

```

write(1, "\320\237\320\276\320\264\321\201\321\207\320\265\321\202 E
\321\204\320\276\321\200\320\274\321\203\320\273\320\276\320"..., 47Подсчет E формулой
(1+1/x)^x...
) = 47
write(1, "\320\236\321\202\320\262\320\265\321\202: 2.694411\n\n", 22Ответ: 2.694411

) = 22
write(1, ": ", 2: ) = 2
read(0, 0
"0\n", 1024) = 2
munmap(0x7fc3949ec000, 16448) = 0
openat(AT_FDCWD, "./librealization2.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"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=19296, ...}) = 0
getcwd("/home/igor/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/c/OC/lab5", 128) = 45
mmap(NULL, 16448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fc3949ec000
mmap(0x7fc3949ed000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fc3949ed000
mmap(0x7fc3949ee000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fc3949ee000
mmap(0x7fc3949ef000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fc3949ef000
close(3) = 0
mprotect(0x7fc3949ef000, 4096, PROT_READ) = 0
write(1, "\320\227\320\260\320\263\321\200\321\203\320\267\320\272\320\260
\320\277\321\200\320\276\321\210\320\273\320\260 \321\203"..., 45Загрузка прошла успешно
) = 45
write(1, "\n", 1
) = 1
write(1, ": ", 2: ) = 2
read(0, 0
"0\n", 1024) = 2
munmap(0x7fc3949ec000, 16448) = 0
openat(AT_FDCWD, "./librealization1.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"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=19240, ...}) = 0
getcwd("/home/igor/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/c/OC/lab5", 128) = 45
mmap(NULL, 16448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fc3949ec000
mmap(0x7fc3949ed000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fc3949ed000
mmap(0x7fc3949ee000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fc3949ee000

```

```

mmap(0x7fc3949ef000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fc3949ef000
close(3) = 0
mprotect(0x7fc3949ef000, 4096, PROT_READ) = 0
write(1, "\320\227\320\260\320\263\321\200\321\203\320\267\320\272\320\260
\320\277\321\200\320\276\321\210\320\273\320\260 \321\203"..., 45Загрузка прошла успешно
) = 45
write(1, "\n", 1
) = 1
write(1, ": ", 2) = 2
read(0, "", 1024) = 0
munmap(0x7fc3949ec000, 16448) = 0
exit_group(0) = ?
+++ exited with 0 +++
igor@igor-Aspire-A315-53G:~/Рабочий стол/с/ОС/lab5$

```

## Выводы

В данной лабораторной работе я научился использовать, создавать и подключать динамические библиотеки. Самым сложным заданием из всей лабораторной работы оказалась линковка библиотек в makefile, из-за чего я специально представил в данной лабораторной работе. Я посмотрел множество видео, снятых на разных языках, и прочитал множество статей, чтобы понять, какие флаги мне использовать и где их ставить. Отсутствие хороших материалов и примеров по данной лабораторной работе сильно усложнило ее выполнение. Я очень долго не понимал, почему выбрасывается ошибка неопределенного символа `sin` и лишь потратив день, я все смог решить данную проблему. Для отладки линковки пришлось применять утилиту `nm`, которая показывает таблицу имен вызываемых функций в бинарном файле.

## Список литературы

1. Поисковик Google [электронный ресурс] URL: <https://google.com/> (дата обращения: 11.12.2020)

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