

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-216Б-23

Студент: Котляр Д.А.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 10.10.24

Москва, 2025

Постановка задачи

Вариант 5.

Требуется создать динамические библиотеки, которые реализуют заданный вариантом 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 информацию, полученную на этапе компиляции;
- Тестовая программа (программа №2), которая загружает библиотеки, используя только их относительные пути и контракты.

Контракты и реализации функций:

1. Расчет интеграла функции $\sin(x)$ на отрезке $[A, B]$ с шагом e :

Сигнатура функции:

`float sin_integral(float a, float b, float e);`

- Реализация №1: Подсчет интеграла методом прямоугольников;
- Реализация №2: Подсчет интеграла методом трапеций.

2. Расчет значения числа e (основание натурального логарифма):

Сигнатура функции: `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:

- `void *dlopen(const char *filename, int flags);` – загружает динамическую библиотеку в память процесса
- `void *dlsym(void *handle, const char *symbol);` – получает адрес функции из загруженной библиотеки
- `int dlclose(void *handle);` – выгружает динамическую библиотеку из памяти
- `ssize_t write(int fd, const void *buf, size_t count);` – выводит сообщение об ошибке в стандартный поток ошибок

В рамках лабораторной работы была реализована система работы с динамическими библиотеками двумя различными подходами. Были разработаны две динамические библиотеки librect.so и libtrap.so, каждая из которых содержит реализации функций sin_integral() и e_func() с разными алгоритмами вычислений: метод прямоугольников и формула $(1 + 1/x)^x$ в первой библиотеке, метод трапеций и сумма ряда $1/n!$ во второй.

Созданы две тестовые программы с единым пользовательским интерфейсом. Первая программа использует статическое связывание с библиотекой на этапе компиляции, где вызовы функций разрешаются при линковке. Вторая программа реализует динамическую загрузку библиотек во время выполнения с помощью системных вызовов dlopen(), dlsym() и dlclose(). Эта программа позволяет переключаться между библиотеками по команде "0" без перезапуска. обе программы поддерживают интерактивный ввод команд.

Код программы

contracts.c

```
#pragma once
```

```
typedef float sin_integral_func(float a, float b, float e);
```

```
typedef float e_func_func(int x);
```

lib_1.c

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

```
#include "contracts.h"
```

```
float sin_integral(float a, float b, float e) {
```

```
    float sum = 0.0f;
```

```
    for (float x = a; x < b; x += e)
```

```
        sum += sinf(x) * e;
```

```
    return sum;
```

```
}
```

```
float e_func(int x) {
```

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

```
}
```

lib_2.c

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

```
#include "contracts.h"
```

```
float sin_integral(float a, float b, float e) {  
    float sum = 0.0f;  
    for (float x = a; x < b; x += e)  
        sum += (sinf(x) + sinf(x + e)) * 0.5f * e;  
    return sum;  
}
```

```
float e_func(int x) {  
    float sum = 1.0f, fact = 1.0f;  
    for (int i = 1; i <= x; i++) {  
        fact *= i;  
        sum += 1.0f / fact;  
    }  
    return sum;  
}
```

main_1.c

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

```
#include "contracts.h"
```

```
float sin_integral(float, float, float);
```

```
float e_func(int);
```

```
int main() {
```

```

int cmd;

while (scanf("%d", &cmd) == 1) {
    if (cmd == 1) {
        float a,b,e;

        scanf("%f%f%f", &a, &b, &e);

        printf("%f\n", sin_integral(a,b,e));
    } else if (cmd == 2) {
        int x;

        scanf("%d", &x);

        printf("%f\n", e_func(x));
    }
}
}

```

main_1.c

```

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <dlfcn.h>

#include "contracts.h"

static sin_integral_func *sin_integral;

static e_func_func *e_func;

static float stub_sin(float a, float b, float e) {
    (void)a; (void)b; (void)e;

    return 0.0f;
}

```

```

static float stub_e(int x) {
    (void)x;
    return 0.0f;
}

void load_library(const char *path, void **lib) {
    if (*lib) dlclose(*lib);

    *lib = dlopen(path, RTLD_LOCAL | RTLD_NOW);
    if (*lib == NULL) {
        write(2, "warning: failed to load library\n", 33);
        sin_integral = stub_sin;
        e_func = stub_e;
        return;
    }

    sin_integral = dlsym(*lib, "sin_integral");
    if (!sin_integral) sin_integral = stub_sin;

    e_func = dlsym(*lib, "e_func");
    if (!e_func) e_func = stub_e;
}

int main() {
    void *lib = NULL;
    int toggle = 0;

    load_library("./librect.so", &lib);

    int cmd;

```

```

while (scanf("%d", &cmd) == 1) {
    if (cmd == 0) {
        toggle = !toggle;
        load_library(toggle ? "./libtrap.so" : "./librect.so", &lib);
    } else if (cmd == 1) {
        float a,b,e;
        scanf("%f %f %f", &a, &b, &e);
        printf("%f\n", sin_integral(a,b,e));
    } else if (cmd == 2) {
        int x;
        scanf("%d", &x);
        printf("%f\n", e_func(x));
    }
}

if (lib) dlclose(lib);
}

```

Протокол работы программы

Тест 1:

```

mai@ManyaBook:/mnt/d/LAB2/OS/laba4$ LD_LIBRARY_PATH=. ./app_static
1
1.2 1.3 1.2
1.118447

```

Тест 2:

```

mai@ManyaBook:/mnt/d/LAB2/OS/laba4$ LD_LIBRARY_PATH=. strace -o output.txt ./app_static
2
3
2.370371

```

Тестирование:

```
$ LD_LIBRARY_PATH=. ./app_static
```

Strace:


```

$ LD_LIBRARY_PATH=. strace -o output.txt ./app_static
execve("./app_static", ["/app_static"], 0x7ffe7c3f4bc0 /* 28 vars */) = 0
brk(NULL)                                = 0x600992fcd000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff3d50be70) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x777cf9e12000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/lib_1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/lib_1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./tls/x86_64/x86_64/lib_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD, "./tls/x86_64/lib_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "./tls/x86_64/lib_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "./tls/lib_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "./x86_64/x86_64/lib_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD, "./x86_64/lib_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD, "./x86_64/lib_1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD, "./lib_1.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"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0777, st_size=15464, ...}, AT_EMPTY_PATH) = 0
getcwd("/mnt/d/LAB2/OS/laba4", 128)     = 21
mmap(NULL, 16440, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x777cf9e0d000
mmap(0x777cf9e0e000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x777cf9e0e000
mmap(0x777cf9e0f000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x777cf9e0f000

```



```

mmap(0x777cf9da1000, 372736, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x777cf9da1000

mmap(0x777cf9dfc000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x777cf9dfc000

close(3) = 0

openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "./tls/x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

openat(AT_FDCWD, "./tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)

openat(AT_FDCWD, "./tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)

openat(AT_FDCWD, "./tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or
directory)

openat(AT_FDCWD, "./x86_64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

openat(AT_FDCWD, "./x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)

openat(AT_FDCWD, "./x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)

openat(AT_FDCWD, "./libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or
directory)

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"..., 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) =
0x777cf9a00000

```

mprotect(0x777cf9a28000, 2023424, PROT_NONE) = 0

mmap(0x777cf9a28000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x777cf9a28000

mmap(0x777cf9bbd000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x777cf9bbd000

mmap(0x777cf9c16000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x777cf9c16000

mmap(0x777cf9c1c000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x777cf9c1c000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x777cf9d14000

arch_prctl(ARCH_SET_FS, 0x777cf9d14740) = 0

set_tid_address(0x777cf9d14a10) = 8941

set_robust_list(0x777cf9d14a20, 24) = 0

rseq(0x777cf9d150e0, 0x20, 0, 0x53053053) = 0

mprotect(0x777cf9c16000, 16384, PROT_READ) = 0

mprotect(0x777cf9dfc000, 4096, PROT_READ) = 0

mprotect(0x777cf9e10000, 4096, PROT_READ) = 0

mprotect(0x60096c9f0000, 4096, PROT_READ) = 0

mprotect(0x777cf9e4c000, 8192, PROT_READ) = 0

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

munmap(0x777cf9dfe000, 59280) = 0

newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH)
= 0

getrandom("\x8f\x7e\xe8\x5c\xf9\x37\xbd\x4a", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x600992fcd000

brk(0x600992fee000) = 0x600992fee000

read(0, "2\n", 1024) = 2

read(0, "3\n", 1024) = 2

newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH)
= 0

```
write(1, "2.370371\n", 9)          = 9
read(0, "\320\262\n", 1024)       = 3
lseek(0, -3, SEEK_CUR)              = -1 ESPIPE (Illegal seek)
exit_group(0)                       = ?
+++ exited with 0 +++
```

Вывод

Программа успешно реализовала два подхода работы с динамическими библиотеками. Были разработаны две библиотеки librect.so и libtrap.so с разными алгоритмами вычислений. Для статического связывания создана программа, использующая библиотеку на этапе компиляции. Для динамической загрузки реализована программа с функцией переключения библиотек во время выполнения. Основные проблемы при выполнении работы возникли с пониманием принципов динамической загрузки библиотек, особенно в области правильного использования функций dlopen(), dlsym() и dlclose().