

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-209БВ-24

Студент: Корепанов Иван Алексеевич

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 12.12.25

Москва, 2025

Постановка задачи

Вариант 27.

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 используют одну из библиотек, используя информацию полученные на этапе компиляции;
- Тестовая программа (*программа №2*), которая загружает библиотеки, используя только их относительные пути и контракты.

№	Описание	Сигнатура	Реализация	Реализация
			1	2
1	Рассчет значения числа Пи при заданной длине ряда (K)	float Pi(int K)	Ряд Лейбница	Формула Валлиса
2	Рассчет значения числа е(основание натурального логарифма)	Float E(int x)	$(1 + 1/x)^x$	Сумма ряда по n от 0 до x, где элементы ряда равны: $(1/(n!))$

Общий метод и алгоритм решения

Использованные системные вызовы

- `void* dlopen(const char* filename, int flag);` - загружает динамическую библиотеку в память
- `int dlclose(void* handle);` - выгружает динамическую библиотеку из памяти
- `void* dlsym(void* handle, const char* symbol);` - получает адрес символа из библиотеки
- `char* dlerror(void);` - возвращает строку с описанием последней ошибки

Алгоритм работы:

Программа представляет собой интерактивное консольное приложение, которое позволяет пользователю выполнять две математические операции:

1. вычисление приближённого значения числа π при заданной длине ряда (K);
2. вычисление приближённого значения числа e при заданном параметре (x).

В динамическом режиме программа также поддерживает переключение между двумя различными реализациями этих функций, выполненными в отдельных динамических библиотеках.

Программа работает в бесконечном цикле, пока пользователь не введёт команду для выхода. Ниже приведено пошаговое описание работы.

- **static** — функции Pi и E подключены на этапе линковки;
- **dynamic** — функции загружаются из библиотек при помощи `dlopen()` и `dlsym()`.

При динамической загрузке определяется активная библиотека, и получаются указатели на функции Pi и E .

Код программы

mathlib.h

```
#ifndef MATHLIB_H
#define MATHLIB_H
float Pi(int K);
float E(int x);
#endif
```

lib1.c

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

```
#include "mathlib.h"

float Pi(int K) {
    float pi = 0.0f;
    for(int n = 0; n < K; n++)
        pi += powf(-1, n) / (2*n + 1);
    return pi * 4;
}

float E(int x) {
    return powf(1.0f + 1.0f/x, x);
}
```

lib2.c

```
#include "mathlib.h"

float Pi(int K) {
    float pi = 1.0f;
    for(int n = 1; n <= K; n++)
        pi *= (4.0f*n*n) / (4.0f*n*n - 1);
    return pi * 2;
}

float E(int x) {
    float e = 1.0f;
    float fact = 1.0f;
    for(int n = 1; n <= x; n++) {
        fact *= n;
        e += 1.0f / fact;
    }
    return e;
}
```

```
}
```

dynamic.c

```
#include <stdio.h>
#include <dlfcn.h>
#include "mathlib.h"

int main() {
    char *libs[] = {"./lib1.so", "./lib2.so"};
    int cur = 0;
    void *handle = dlopen(libs[cur], RTLD_LAZY);
    float (*PiFunc)(int) = dlsym(handle, "Pi");
    float (*EFunc)(int) = dlsym(handle, "E");
    int cmd, arg;
    while (scanf("%d", &cmd) != EOF) {
        if (cmd == 0) {
            dlclose(handle);
            cur = 1 - cur;
            handle = dlopen(libs[cur], RTLD_LAZY);
            PiFunc = dlsym(handle, "Pi");
            EFunc = dlsym(handle, "E");
            printf("свалиута\n");
        }
        else if (cmd == 1) {
            scanf("%d", &arg);
            printf("Pi = %f\n", PiFunc(arg));
        }
        else if (cmd == 2) {
            scanf("%d", &arg);
        }
    }
}
```

```
    printf("E = %f\n", EFunc(arg));  
}  
}  
dlclose(handle);  
return 0;  
}
```

static.c

```
#include <stdio.h>  
  
#include "mathlib.h"  
  
  
int main() {  
    int cmd, arg;  
    while (scanf("%d", &cmd) != EOF) {  
        if (cmd == 1) {  
            scanf("%d", &arg);  
            printf("Pi = %f\n", Pi(arg));  
        }  
        else if (cmd == 2) {  
            scanf("%d", &arg);  
            printf("E = %f\n", E(arg));  
        }  
    }  
    return 0;  
}
```

Протокол работы программы

Тестирование:

vscode → /workspaces/OS_LABS/lab4/src (main) \$./static

Pi = 3.131593

2 100

E = 2.704811

vscode → /workspaces/OS_LABS/lab4/src (main) \$./dynamic

1 100

Pi = 3.131593

2 100

E = 2.704811

0

свалнута

1 100

Pi = 3.133788

2 100

E = 2.718282

Strace:

Static.c

vscode → /workspaces/OS_LABS/lab4/src (main) \$ strace -f ./static

execve("./static", ["/./static"], 0x7ffe58c101c8 /* 33 vars */) = 0

brk(NULL) = 0x6096b9ea2000

access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./tls/x86_64/x86_64/lib1.so", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "./tls/x86_64/lib1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "./tls/x86_64/lib1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "./tls/lib1.so", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

openat(AT_FDCWD, "./x86_64/x86_64/lib1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "./x86_64/lib1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "./x86_64/lib1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

```
openat(AT_FDCWD, "./lib1.so", O_RDONLY|O_CLOEXEC) = 3
```

`fstat(3, {st_mode=S_IFREG|0755, st_size=15992, ...}) = 0`

```
mmap(NULL, 8192, PROT_READ|PROT_WRITE,  
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7e066cb55000
```

```
getcwd("/workspaces/OS_LABS/lab4/src", 128) = 29
```

```
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE,  
3, 0) = 0x7e066cb50000
```

```
mmap(0x7e066cb51000, 4096, PROT_READ|PROT_EXEC,  
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7e066cb51000
```

mmap(0x7e066cb52000, 4096, PROT_READ,

MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7e066cb52000

`mmap(0x7e066cb53000, 8192, PROT_READ|PROT_WRITE,`

MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7e066cb53000

close(3) = 0

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)


```
mmap(0x7e066c974000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x141000) = 0x7e066c974000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7e066c82f000
arch_prctl(ARCH_SET_FS, 0x7e066c82f740) = 0
mprotect(0x7e066cb40000, 16384, PROT_READ) = 0
mprotect(0x7e066c974000, 4096, PROT_READ) = 0
mprotect(0x7e066cb53000, 4096, PROT_READ) = 0
mprotect(0x6096a45f7000, 4096, PROT_READ) = 0
mprotect(0x7e066cb81000, 4096, PROT_READ) = 0
munmap(0x7e066cb4a000, 21784) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
brk(NULL) = 0x6096b9ea2000
brk(0x6096b9ec3000) = 0x6096b9ec3000
read(0, 1 100
"1 100\n", 1024) = 6
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
write(1, "Pi = 3.131593\n", 14Pi = 3.131593
) = 14
read(0, 2 100
"2 100\n", 1024) = 6
write(1, "E = 2.704811\n", 13E = 2.704811
) = 13
read(0, -1
"-1\n", 1024) = 3
write(1, "Exit\n", 5Exit
) = 5
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++
```

Dynamic.c

Вывод

В ходе выполнения лабораторной работы была реализована программа с поддержкой статической и динамической загрузки функций через разделяемые библиотеки. Были созданы две библиотечные реализации и организовано

переключение между ними в процессе выполнения программы. Данные программы демонстрируют различия между статистической и динамической линковкой.