

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский авиационный институт
(Национальный исследовательский университет)»

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

ОТЧЁТ
по лабораторной работе № 4
по дисциплине «Операционные системы»

Студент: Князьков Николай Дмитриевич
Группа: М80–203БВ–24
Вариант: 29
Преподаватель: Соколов А. А.

Москва, 2025

Лабораторная работа № 4

1.1 Постановка задачи

Цель работы: освоение создания динамических библиотек и использования функций из них в программах.

Задание: для заданного варианта реализовать в динамических библиотеках две математические функции (Pi и E) с двумя реализациями каждой. Написать две тестовые программы: одна со статической линковкой, другая с динамической загрузкой библиотек во время исполнения. Пользователь может переключаться между реализациями функций.

Вариант 29: - Функция $\text{Pi}(K)$ – вычисление числа π с длиной ряда K (ряд Лейбница и формула Валлиса). - Функция $\text{E}(x)$ – вычисление экспоненты e^x (реализации: $(1 + 1/x)^x$ и сумма ряда).

Команды пользователя:

- 0 – переключить между реализациями в динамической программе;
- 1 arg1 – вызвать функцию Pi с аргументом arg1 и вывести результат;
- 2 arg1 – вызвать функцию E с аргументом arg1 и вывести результат.

1.2 Общие сведения о программе

Созданы четыре динамические библиотеки: `libpi_leibniz.so`, `libpi_wallis.so`, `libe_powers.so`, `libe_series.so`, каждая содержит одну реализацию функции.

Программы:

- `test_static.c` – статическая линковка с одной из библиотек;
- `test_dynamic.c` – динамическая загрузка библиотек через `dlopen()` и получение адресов функций через `dlsym()`.

Заголовочный файл `math_contracts.h` содержит объявления функций:

```
1 float Pi(int K);
2 float E(int x);
```

1.3 Метод и алгоритм решения

Функции реализованы следующим образом:

- Pi (Лейбница): $Pi = 4 * (1 - 1/3 + 1/5 - 1/7 + \dots)$, K терминов;
- Pi (Валлиса): $Pi = 2 * (2/1 * 2/3 * 4/3 * 4/5 \dots)$, K множителей;
- $E(1)$: $(1 + 1/x)^x$;
- $E(2)$: $E = 1 + 1/1! + 1/2! + \dots + 1/x!$.

В статической программе вызывается одна реализация. В динамической программе при вводе команды 0 текущая библиотека освобождается (`dlclose()`) и загружается другая (`dlopen()`). Аргументы команд парсятся и передаются в функции `Pi()` и `E()`.

1.4 Основные файлы программы с кодом

Пример динамической загрузки функций (`test_dynamic.c`):

```
1 #include <stdio.h>
2 #include <dlfcn.h>
3 #include "math_contracts.h"
4
5 int main() {
6     void *libpi = dlopen("libpi_leibniz.so", RTLD_LAZY);
7     void *libe = dlopen("libe_powers.so", RTLD_LAZY);
8     float (*Pi)(int) = dlsym(libpi, "Pi");
9     float (*E)(int) = dlsym(libe, "E");
10    int cmd;
11    int libpi_leibniz = 1, libe_powers = 1;
12    while(scanf("%d", &cmd) == 1) {
13        if(cmd == 0) {
14            //                                         Pi
15            dlclose(libpi);
16            if(libpi_leibniz) {
17                libpi = dlopen("libpi_wallis.so", RTLD_LAZY);
18                libpi_leibniz = 0;
19            } else {
20                libpi = dlopen("libpi_leibniz.so", RTLD_LAZY);
21                libpi_leibniz = 1;
```

```

22     }
23     Pi = dlsym(libpi, "Pi");
24     //                                         E
25     dlclose(libe);
26     if(libe_powers) {
27         libe = dlopen("libe_series.so", RTLD_LAZY);
28         libe_powers = 0;
29     } else {
30         libe = dlopen("libe_powers.so", RTLD_LAZY);
31         libe_powers = 1;
32     }
33     E = dlsym(libe, "E");
34 }
35 else if(cmd == 1) {
36     int k; scanf("%d", &k);
37     printf("Pi(%d) = %f\n", k, Pi(k));
38 }
39 else if(cmd == 2) {
40     int x; scanf("%d", &x);
41     printf("E(%d) = %f\n", x, E(x));
42 }
43 }
44 return 0;
45 }
```

Листинг 1.1: *test_{dynamic.c}*

Пример реализации Ри через ряд Лейбница (*pi_leibniz.c*):

```

1 #include "math_contracts.h"
2 float Pi(int K) {
3     float sum = 0.0f;
4     for(int i = 0; i < K; i++) {
5         sum += (i % 2 == 0 ? 1.0f : -1.0f) / (2*i + 1);
6     }
7     return 4.0f * sum;
8 }
```

Листинг 1.2: *pi_leibniz.c*

1.5 Пример работы

Статическая программа:

```
1 10000 // Pi(10000)
Pi(10000) = 3.141492
2 10 // E(10)
E(10) = 2.718282
```

Динамическая программа:

```
0 // переключаем Pi и E на другую реализацию
1 10000 // вызываем Pi(10000) - теперь формула Валлиса
Pi(10000) = 3.141593
0 // переключаем обратно
2 1000 // вызываем E(1000) - теперь сумма ряда
E(1000) = 2.716923
```

1.6 Выводы

В лабораторной работе разработаны динамические библиотеки для вычисления числа π и экспоненты двумя методами. Продемонстрированы статическая линковка и динамическая загрузка библиотек. Программа корректно обрабатывает команды пользователя, переключение реализаций работает, функции возвращают ожидаемые значения. Проанализированы различия методов и затраты вычислительных ресурсов.