

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №5 по курсу
«Операционные системы»

Тема работы
“Динамические библиотеки”

Студент: Андреев Александр Олегович
Группа: М8О-206Б-20
Вариант: 20
Преподаватель: Соколов Андрей Алексеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

Содержание

1. Постановка задачи
2. Исходный код
3. Демонстрация работы программы
4. Выводы

Постановка задачи

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 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 знания полученные на этапе компиляции;
- 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 только для программы №2). Можно реализовать лабораторную работу без данной функции, но максимальная оценка в этом случае будет «хорошо»;
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.

Исходный код

prog1.c

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

```
void Usage(){
    printf("1. For searching pi. Input is int number of iterations\n2. For searching e.
    Input is int number of iterations\n");
}
```

```
int main() {
    int cmd;
    printf("Enter number the command and args: ");
    while (scanf("%d", &cmd) == 1) {
        if (cmd == 1) {
            int n;
            if (scanf("%d", &n) != 1) {
                fprintf(stderr, "Function piSearch error input\n");
                exit(2);
            }
            printf("pi: %lf\n", piSearch(n));
        }
    }
}
```

```

else if (cmd == 2) {
    int n;
    if (scanf("%d", &n) != 1) {
        fprintf(stderr, "Function eSearch error input\n");
        exit(1);
    }
    printf("e = %lf", eSearch(n));
}
else {
    fprintf(stderr, "Undefined command\n");
    Usage();
}
printf("Enter number the command and args: \n");
}
return 0;
}

```

lib1.c

```

#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <math.h>
#include "funcs.h"

```

```

double piSearch(int n) {
    printf("This is pi_leibniz realise of pi\n");
    double mpi = 0;
    double zn = 1.0;
    for (int i = 0; i < n; zn += 2.0, i++) {
        if (i % 2) {

```

```

        mpi -= 1.0 / zn;
    }
    else {
        mpi += 1.0 / zn;
    }
}
return mpi * 4.0;
}

```

```

double eSearch (int n) {
    if (n < 0) {
        return -1;
    }
    double e = 1.0;
    for (int i = 0; i < n; ++i) {
        e *= 1 + 1/(double)n;
    }
    return e;
}

```

lib2.c

```

#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <math.h>
#include "funcs.h"

```

```

double piSearch(int n) {
    printf("This is vallis realise of pi\n");
    double mpi = 1.0;

```

```

    for (int i = 1; i <= n; i++) {
        mpi *= (4.0 * (double)i * (double)i) / (4.0 * (double)i * (double)i - 1.0);
    }
    return mpi * 2.0;
}

```

```

double eSearch(int n) {
    if (n < 0) {
        return -1;
    }
    double e = 1.0;
    double term = 1.0;
    for (int i = 1; i <= n; ++i) {
        term /= i;
        e += term;
    }
    return e;
}

```

funcs.h

```

#ifndef FUNCS_H
#define FUNCS_H

```

```

double piSearch(int n);

```

```

double eSearch(int n);

```

```

#endif

```

prog2.c

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

```
Usage(){
    printf("0.Change lib.\n");
    printf("1. For searching pi.\n");
    printf("2. For searching e.\n");
}
```

```
int main() {
    double (*piSearch)(int);
    double (*eSearch)(int);
    void* lib1 = dlopen("lib1.so", RTLD_LAZY);
    if (!lib1) {
        fprintf(stderr, "Cannot open library lib.so\n");
        return -2;
    }
    void* lib2 = dlopen("lib2.so", RTLD_LAZY);
    if (!lib2) {
        fprintf(stderr, "Cannot open library lib.so\n");
        return -2;
    }
    piSearch = dlsym(lib1, "piSearch");
    eSearch = dlsym(lib1, "eSearch");
    int cmd;
    bool first_realise = false;
    Usage();
    printf("Enter number the command and args: ");
```



```

while (scanf("%d", &cmd) == 1) {
    if (cmd == 0) {
        if (first_realise) {
            piSearch = dlsym(lib1, "piSearch");
            eSearch = dlsym(lib1, "eSearch");
        }
        else {
            piSearch = dlsym(lib2, "piSearch");
            eSearch = dlsym(lib2, "eSearch");
        }
        first_realise = !first_realise;
        if (piSearch == NULL || eSearch == NULL) {
            fprintf(stderr, "Cannot find functions\n");
            exit(2);
        }
    }
    else if (cmd == 1) {
        int n;
        if (scanf("%d", &n) != 1) {
            fprintf(stderr, "Function piSearch error input\n");
            exit(1);
        }
        printf("pi = %lf\n", piSearch(n));
    }
    else if (cmd == 2) {
        int n;
        if (scanf("%d", &n) != 1) {
            fprintf(stderr, "Function eSearch error input\n");
            exit(1);
        }
    }
}

```

```

        printf("e = %lf\n", eSearch(n));
    }
    else {
        fprintf(stderr, "Undefined command\n");
        Usage();
    }
    printf("Enter number the command and args: ");
}
dlclose(lib1);
dlclose(lib2);
return 0;
}

```

Демонстрация работы программы

Тест 1. Первый вариант программы.

missclick3@missclick3:~/Desktop/OSLabs/lab5/l5Static\$./a.out

Enter number the command and args: 1

100

This is pi_leibniz realise of pi

pi: 3.131593

Enter number the command and args:

2

10000

e = 2.718146Enter number the command and args:

Тест 2. Второй вариант работы программы.

missclick3@missclick3:~/Desktop/OSLabs/lab5/l5Dynamic\$./prog2

0.Change lib.

1. For searching pi.

2. For searching e.

Enter number the command and args: 1

10000

This is pi_leibniz realise of pi

$\pi = 3.141493$

Enter number the command and args: 0

Enter number the command and args: 1

19191991

This is vallis realise of pi

$\pi = 3.141593$

Выводы

В ходе данной лабораторной работы я познакомился с несколькими способами подключения библиотек. Особенно хотелось бы выделить динамическое подключение, которое происходит без подставления кода в саму программу.