

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: Каширин Кирилл Дмитриевич

Группа: М8О-208Б-20

Вариант: 5

Преподаватель: Миронов Евгений Сергеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2021

Содержание

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

Репозиторий

<https://github.com/lumses/OS>

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).
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.

Общие сведения о программе

Программа написана на языке Си в UNIX-подобной операционной системе.

Сборка проекта происходит при помощи make-файла

```
gcc -fPIC -c lib1.c -o d1.o
```

```
gcc -fPIC -c lib2.c -o d2.o
```

```
gcc -shared -o libd1.so d1.o -lm
```

```
gcc -shared -o libd2.so d2.o -lm
```

```
gcc programm1.c -L. -ld1 -o main1 -Wl,-rpath -Wl,. -lm
```

```
gcc programm2.c -L. -ldl -o main2 -Wl,-rpath -Wl,. -lm
```

Флаг `-fPIC` означает, что используется относительная переадресация в переходах подпрограмм.

Флаг `-shared` предписывается, что создается динамическая библиотека.

Флаг `-L.` - ищет в каталоге файлы библиотеки.

Флаг `-ldl` используется для работ с функциями `dlopen`, `dlsym`, `dlclose`.

Флаг `-lm/-ld1` связывает с файлом библиотеку `math/lib1`.

Общий метод и алгоритм решения

В файлах `lib1.c` и `lib2.c` реализовал функции методами, соответствующие условию задания. В тестовой программе №1, которая использует данные библиотеки во время компиляции, прописал цикл `while`, в условии которого идет чтение ключа. Пока ключ может считаться, в зависимости от его значения я выбираю функцию, которая будет обрабатываться и выводить результат в консоль (1 – `SinIntegral`, 2- `E`). После выбора функции считываю входные данные для функции, вычисляю её и вывожу результат.

В тестовой программе №2 изначально прошу пользователя выбрать стартовую библиотеку. В зависимости от его выбора, загружаю открываю ту или иную библиотеку с помощью функции `dlopen`. Нахожу начало адреса функций `SinIntegral` и `E` с помощью функции `dlsym`. В цикле `while` считываю ключ и в зависимости от его значения обрабатываю ту или иную функцию и вывожу результат на консоль. Если ключ равен 0, с помощью функции `dlclose` закрываю библиотеку и открываю, и загружаю (`dlopen`) другую библиотеку. Также с помощью `dlsym` нахожу начало адреса функций `SinIntegral` и `E` уже другой библиотеки. После окончания ввода, закрываю библиотеку.

Исходный код

lib1.c

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

```
float SinIntegral(float A, float B, float e){  
    float integral = 0;  
    for(float i = A; i < B; i+=e ){  
        integral += sin(i) * e;  
    }  
    return integral;  
}
```

```
float E(int x) {  
    return pow(1 + 1. / x, x);  
}
```

lib2.c

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

```
float SinIntegral( float A, float B, float e){  
    float integral = 0;  
    for(float i = A + e; i < B; i+=e ){  
        integral += (sin(i) + sin(i - e)) / 2 * e;  
    }  
    return integral;  
}
```

```
float E(int x) {  
    float answer = 0;  
    long long s = 1;  
    for (int i = 0; i <= x; ++i) {  
        if (i!=0) {  
            s*=i;  
        }  
        answer += 1. / s;  
    }  
    return answer;  
}
```

programm1.c

```
#include<stdio.h>

float SinIntegral(float A,float B,float e);
float E(int x);

int main(){
    int key;
    while(scanf("%d",&key)>0){
        if (key == 1) {
            float A,B,e;
            scanf("%f%f%f",&A,&B,&e);
            printf("%s(%f, %f, %f) = %f\n", "SinIntegral", A,B,e, SinIntegral(A,B,e));
        } else if (key == 2) {
            int x;
            scanf("%d",&x);
            printf("%s(%d) = %f\n", "E", x, E(x));
        } else {
            printf("ERROR");
        }
    }
}
```

programm2.c

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

int main(int argc, char* argv[]){

    void *ext_library = NULL;
    float (*SinIntegral)(float A, float B, float e);
    float (*E)(int x);
    //загрузка библиотеки
    int key; int start_library;
    printf("Enter start library: 1 or 2?");
    scanf("%d",&start_library);
```

```

if (start_library == 1) {
    ext_library =
dlopen("/home/kirill/OS/os_lab5/libd1.so",RTLD_LAZY);
} else if (start_library == 2) {
    ext_library =
dlopen("/home/kirill/OS/os_lab5/libd2.so",RTLD_LAZY);
} else {
    printf("You don't write start key\n");
}
if (!ext_library){
    fprintf(stderr,"dlopen() error: %s\n", dlerror());
    return 1;
}
SinIntegral = dlsym(ext_library,"SinIntegral");
E = dlsym(ext_library, "E");
while (scanf("%d",&key) > 0 ) {
    if (key == 1) {
        float A,B,e;
        scanf("%f%f%f",&A,&B,&e);
        printf("%s(%f, %f, %f) =
%f\n","SinIntegral",A,B,e,(*SinIntegral)(A,B,e));
    } else if (key == 2) {
        int x;
        scanf("%d",&x);
        printf("%s(%d) = %f\n","E",x,(*E)(x));
    } else if (key == 0) {
        dlclose(ext_library);
        if (start_library == 1) {
            ext_library = dlopen("./libd2.so",RTLD_LAZY);
            start_library = 2;
        } else if (start_library == 2) {
            ext_library = dlopen("./libd1.so",RTLD_LAZY);
            start_library = 1;
        }
        if (!ext_library) {
            fprintf(stderr,"dlopen() error: %s\n",
dlerror());
            return 1;
        }
        SinIntegral = dlsym(ext_library,"SinIntegral");

```

```

        E = dlsym(ext_library, "E");
    }
}
dlclose(ext_library);
}

```

Демонстрация работы программы

```

kirill@LAPTOP-F153AKTP:~$ cd OS/os_lab5/src
kirill@LAPTOP-F153AKTP:~/OS/os_lab5/src$ make
gcc -fPIC -c lib1.c -o d1.o
gcc -fPIC -c lib2.c -o d2.o
gcc -shared -o libd1.so d1.o -lm
gcc -shared -o libd2.so d2.o -lm
gcc programm1.c -L. -ld1 -o main1 -Wl,-rpath -Wl,. -lm
gcc programm2.c -L. -ld1 -o main2 -Wl,-rpath -Wl,. -lm
kirill@LAPTOP-F153AKTP:~/OS/os_lab5/src$ ./main1
1 0 3.1415 0.001
SinIntegral(0.000000, 3.141500, 0.001000) = 2.000002
2 3
E(3) = 2.370370
kirill@LAPTOP-F153AKTP:~/OS/os_lab5/src$ ./main2
Enter start library: 1 or 2?2
1 0 3.1415 0.001
SinIntegral(0.000000, 3.141500, 0.001000) = 2.000006
1 1.57 2.34 0.2
SinIntegral(1.570000, 2.340000, 0.200000) = 0.562898
2 4
E(4) = 2.708333
2 3
E(3) = 2.666667
0
2 3
E(3) = 2.370370
2 4
E(4) = 2.441406
1 0 3.1415 0.001
SinIntegral(0.000000, 3.141500, 0.001000) = 2.000002
0
2 3
E(3) = 2.666667
kirill@LAPTOP-F153AKTP:~/OS/os_lab5/src$ 

```


Выводы

Эта лабораторная работа познакомила и научила меня работать с динамическими библиотеками. Я описал динамические библиотеки функций подсчет интеграла функции $\sin(x)$ и вычисления числа e . Также я поработал с теорией, понял основной принцип динамических библиотек, их различие от статических библиотек.