

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”

Студентка: Клитная Анастасия Викторовна

Группа: М8О-208Б-20

Вариант: 1

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/klitnaya/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

```
g++ -fPIC -c library1.cpp -o l1.o
```

```
g++ -fPIC -c library2.cpp -o l2.o
```

```
g++ -shared l1.o -o libd1.so
g++ -shared l2.o -o libd2.so
g++ prog2.cpp -L. -ld1 -o pr2 -Wl,-rpath -Wl,.
g++ prog1.cpp -L. -ld1 -o pr1 -Wl,-rpath -Wl,.
```

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 `library1.c` и `library2.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-`Derivative`). 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 `Derivative` 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 `Derivative` уже другой библиотеки. После окончания ввода, закрываю библиотеку.

Исходный код

libary1.cpp

```
#include <cmath>

extern "C" float SinIntegral(float A, float B, float e);
extern "C" float Derivative(float A, float deltaX);

float SinIntegral(float A, float B, float e){
    float Integral = 0;
    for(float i = A; i<B; i+=e){
        Integral += sinf(i) *e;
    }
    return Integral;
}

float Derivative(float A, float deltaX){
    return (cosf (A + deltaX) - cosf(A))/deltaX;
}
```

libary2.cpp

```
#include <cmath>

extern "C" float SinIntegral(float A, float B, float e);
extern "C" float Derivative(float A, float deltaX);

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

```
float Derivative(float A, float deltaX) {
    return (cosf(A + deltaX) - cosf(A - deltaX)) / (2 * deltaX);
}
```

prog1.cpp

```
#include <iostream>
```

```
using namespace std;
```

```
//extern "C" для использования float в работе с C++
```

```
extern "C" float SinIntegral(float A, float B, float e);
```

```
extern "C" float Derivative(float A, float deltaX);
```

```
int main()
```

```
{
```

```
    int flag;
```

```
    cout << "Please enter the flag:\n";
```

```
    while (scanf("%d",&flag) > 0 ){
```

```
        if (flag == 1){
```

```
            cout<<"Please enter your dates:\n";
```

```
            float A, B, e;
```

```
            cin>>A>>B>>e;
```

```
            cout<<"SinIntegral("<< A<<","<<B<<","<< e<<")="<<SinIntegral(A, B, e)<< endl;
```

```
        }
```

```
        else if (flag == 2){
```

```
            cout<<"Please enter your dates:\n";
```

```
            float A, deltaX;
```

```
            cin>>A>>deltaX;
```

```
            cout<< "Derivative("<<A<<","<<deltaX<<")="<<Derivative(A,deltaX)<<endl;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

prog2.cpp

```
#include <stdio.h>

#include <iostream>

#include <stdlib.h>

#include <dlfcn.h>

using namespace std;

int main()
{
    void* adres = NULL; //адрес для доступа к библиотеке
    //указатели на функции 1 и 2
    float (*SinIntegral)(float A, float B, float e);
    float (*Derivative)(float A, float deltaX);

    const char* libeary_mas[]={ "libd1.so", "libd2.so"};
    int curlib;
    int statl;

    cout<<"Enter the start library:"<<endl;
    cout << '\t' << "1 for first library" <<endl;
    cout << '\t' << "2 for second library" <<endl;
    cin>>statl;
    bool flag = 1;
    while(flag){
        if (statl==1){
            curlib = 0;
            flag = 0;
        }
        else if(statl==2){
            curlib=1;
            flag=0;
        }
    }
```

```

else{
    cout<<"You enter the wrong number. Enter again!"<<endl;
    cin>>statl;
}
}

adres = dlopen(libeary_mas[curlib],RTLD_LAZY);//RTLD_LAZY выполняется поиск
только тех символов, на которые есть ссылки из кода

if (!adres){
    cout<<"Error";
    exit(EXIT_FAILURE);
}

SinIntegral = (float (*)(float, float, float))dlsym(adres,"SinIntegral");//возвращаем адрес
функции из памяти библиотеки /dlsym присваивает указателю на функцию, объявленному
в начале, ее адрес в библиотеке

Derivative = (float (*)(float, float))dlsym(adres, "Derivative");
int command;

cout<<"Please read the nexst ruls? before you enter the command";

cout << '\t' << "0 for changing the contract;" << std:: endl;
cout << '\t' << "1 for calculating the SinIntegral; " << std:: endl;
cout << '\t' << "2 for calculating the Derivative " << std:: endl;
while(sprintf("Please enter your command: ") && (scanf("%d", &command)) != EOF){
    if (command==0){
        dlclose(adres);//освобождает указатель на библиотеку и программа перестает ей
пользоваться
        if (curlib==0){
            curlib=1;
            adres=dlopen(libeary_mas[curlib],RTLD_LAZY);
            if(!adres){
                cout<<"Error";
                exit(EXIT_FAILURE);
            }

            SinIntegral = (float (*)(float, float, float))dlsym(adres,"SinIntegral");//возвращаем
адрес функции из памяти библиотеки /dlsym присваивает указателю на функцию,
объявленному в начале, ее адрес в библиотеке

```



```

    Derivative = (float*)(float, float))dlsym(adres, "Derivative");

}

else if(curlib==1){
    curlib=0;
    adres = dlopen(libeary_mas[curlib],RTLD_LAZY);
    if(!adres){
        cout<<"Error";
        exit(EXIT_FAILURE);
    }

    SinIntegral = (float*)(float, float, float))dlsym(adres,"SinIntegral");//возвращаем
адрес функции из памяти библиотеки /dlsym присваивает указателю на функцию,
объявленному в начале, ее адрес в библиотеке

    Derivative = (float*)(float, float))dlsym(adres, "Derivative");

}

cout << "You have changed contracts!" << std:: endl;
}

else if (command==1){
    float A, B, e;
    cin>>A>>B>>e;
    float sinintegral =SinIntegral(A,B,e);
    if (sinintegral==-1){
        cout<<"please enter again\n";
    }
    else{
        cout<<"SinIntegral("<< A<<","<<B<<","<< e<<")="<<SinIntegral(A, B, e)<< endl;
    }
}

else if (command==2){
    float A, deltaX;
    cin>>A>>deltaX;

```

```

float derivative =Derivative(A, deltaX);
if (derivative==-1){
    cout<<"please enter again\n";
}
else{
    cout<< "Derivative("<<A<<","<<deltaX<<")="<<Derivative(A,deltaX)<<endl;
}
}
else{
    cout<<"You had to enter only 0, 1 or 2!" << std:: endl;
}
}
}
dlclose(adres);
return 0;
}

```

Демонстрация работы программы

root@Owl:/mnt/c/Users/Настя# ./m1

Please enter the flag:

1

Please enter your dates:

0 3.1415 0.01

SinIntegral(0,3.1415,0.01)=1.99999

2

Please enter your dates:

0 0.01

Derivative(0,0.01)=-0.00500083

root@Owl:/mnt/c/Users/Настя#

root@Owl:/mnt/c/Users/Настя# ./m2

Enter the start library:

1 for first library

2 for second library

1

Please read the next rules? before you enter the command 0 for changing the contract;

1 for calculating the SinIntegral;

2 for calculating the Derivative;

Please enter your command: 1

0 3.1415 0.01

SinIntegral(0,3.1415,0.01)=1.99999

Please enter your command: 2

0 0.01

Derivative(0,0.01)=-0.00500083

Please enter your command: root@Owl:/mnt/c/Users/Настя#

Выводы

Эта лабораторная работа познакомила и научила меня работать с динамическими библиотеками. Я создала динамические библиотеки функций подсчет интеграла функции $\sin(x)$ и вычисления производную от $\cos(x)$. Также я поработала с теорией, поняла основной принцип динамических библиотек, их различие от статических библиотек.