

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: Ханнанов Руслан Маратович

Группа: М8О-208Б-20

Вариант: 12

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/Naksen/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:

```

1  .PHONY: all clean
2
3  all:
4      gcc -fPIC -c lib1.c -o d1.o
5      gcc -fPIC -c lib2.c -o d2.o
6      gcc -shared -o libd1.so d1.o -lm
7      gcc -shared -o libd2.so d2.o -lm
8      gcc prog1.c -L. -ld1 -o main_1 -Wl,-rpath -Wl,. -lm
9      gcc prog2.c -L. -ld1 -o main_2 -Wl,-rpath -Wl,. -lm
10
11 clean:
12     rm d1.o d2.o libd1.so libd2.so main_1 main_2

```

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 `-ld1` 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 – `Derivative`, 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 `Derivative` 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 `dclose`

закрываю библиотеку и открываю, и загружаю (dlopen) другую библиотеку. Также с помощью dlsym нахожу начало адреса функций Derivative и E уже другой библиотеки. После окончания ввода, закрываю библиотеку.

Исходный код

lib1.c

```
1  #include <math.h>
2
3  float E(int x) {
4      return pow(1 + 1. / x, x);
5  }
6
7  float Derivative(float A, float deltaX) {
8      float ans;
9      ans = (cos(A + deltaX) - cos(A)) / deltaX;
10     return ans;
11 }
```

lib2.c

```
1  #include <math.h>
2
3  float E(int x) {
4      float answer = 0;
5      long long s = 1;
6      for (int i = 0; i <= x; ++i) {
7          if (i!=0) {
8              s*=i;
9          }
10         answer += 1. / s;
11     }
12     return answer;
13 }
14
15 float Derivative(float A, float deltaX) {
16     float ans;
17     ans = (cos(A + deltaX) - cos(A - deltaX)) / (2 * deltaX);
18     return ans;
19 }
```

prog1.c

```
1  #include<stdio.h>
2
3  float E(int x);
4  float Derivative(float A, float deltaX);
5
6  int main(){
7
8      int key;
9      while(scanf("%d", &key) > 0){
10
11          if (key == 1) {
12
13              float A, deltaX;
14              scanf("%f%f", &A, &deltaX);
15              printf("%s(%f, %f) = %f\n", "Derivative", A, deltaX, Derivative(A, deltaX));
16          }
17          else if (key == 2) {
18
19              int x;
20              scanf("%d", &x);
21              printf("%s(%d) = %f\n", "E", x, E(x));
22          }
23          else {
24
25              printf("ERROR\n");
26          }
27      }
28  }
29
30
31 }
```

prog2.c

```

1  #include <stdio.h>
2  #include <dlfcn.h>
3
4  int main(){
5      void* cur_lib = NULL;
6      float (*Derivative)(float A, float deltaX);
7      float (*E)(int x);
8
9      //Загрузка библиотек:
10     int key;
11     int lib_num;
12     printf("Library: [1/2]?\n");
13     scanf("%d",&lib_num);
14
15     //Открываем нужную библиотеку:
16     if (lib_num == 1) {
17
18         cur_lib = dlopen("/home/naksan/cprog/5_lab_OS/libd1.so",RTLD_LAZY);
19
20     } else if (lib_num == 2) {
21
22         cur_lib = dlopen("/home/naksan/cprog/5_lab_OS/libd2.so",RTLD_LAZY);
23
24     } else {
25
26         printf("You don't write start key\n");
27
28     }
29
30     //Нахождения адрессов функций:
31     Derivative = dlsym(cur_lib, "Derivative");
32     E = dlsym(cur_lib, "E");

```

```

33
34 while (scanf("%d",&key) > 0 ) {
35
36     if (key == 1) {
37
38         float A,deltaX;
39         scanf("%f%f",&A,&deltaX);
40         printf("%s(%f, %f) = %f\n","Derivative", A, deltaX,(*Derivative)(A, deltaX));
41
42     } else if (key == 2) {
43
44         int x;
45         scanf("%d",&x);
46         printf("%s(%d) = %f\n","E",x,(*E)(x));
47
48     } else if (key == 0) {
49
50         dlclose(cur_lib);
51
52         if (lib_num == 1) {
53
54             cur_lib = dlopen("./libd2.so",RTLD_LAZY);
55             lib_num = 2;
56
57         } else if (lib_num == 2) {
58
59             cur_lib = dlopen("./libd1.so",RTLD_LAZY);
60             lib_num = 1;
61
62         }
63
64         Derivative = dlsym(cur_lib,"Derivative");
65         E = dlsym(cur_lib, "E");
66
67     }
68
69 }
70
71 dlclose(cur_lib);
72 }

```

Демонстрация работы программы


```

naksan@LAPTOP-TL9L61MA:~/cprog/5_lab_OS$ 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 prog1.c -L. -ld1 -o main_1 -Wl,-rpath -Wl,. -lm
gcc prog2.c -L. -ld1 -o main_2 -Wl,-rpath -Wl,. -lm
naksan@LAPTOP-TL9L61MA:~/cprog/5_lab_OS$ ./main_1
1
3 0.01
Derivative(3.000000, 0.010000) = -0.136168
2
10
E(10) = 2.593742
naksan@LAPTOP-TL9L61MA:~/cprog/5_lab_OS$ ./main_2
Library: [1/2]?
1
1
3 0.01
Derivative(3.000000, 0.010000) = -0.136168
2
10
E(10) = 2.593742
3
0
1
3 0.01
Derivative(3.000000, 0.010000) = -0.141118
2
10
E(10) = 2.718282
naksan@LAPTOP-TL9L61MA:~/cprog/5_lab_OS$ █

```

Выводы

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