

**Московский авиационный институт
(национальный исследовательский университет)**

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Операционные системы»

Студент: Э. Л. Носов
Преподаватель: Е. С. Миронов
Группа: М8О-307Б
Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №2

Цель работы: Целью является приобретение практических навыков в:

- Создание динамических библиотек
- Создание программ, которые используют функции динамических библиотек

Задание: 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 №32:

6	Расчет значения числа e (основание натурального логарифма)	Float E(int x)	$(1 + 1/x)^x$	Сумма ряда по n от 0 до x , где элементы ряда равны: $(1/(n!))$
8	Перевод числа x из десятичной системы счисления в другую	Char* translation(long x)	Другая система счисления двоичная	Другая система счисления троичная

1 Описание

В файлах `lib1.c` и `lib2.c` написаны библиотеки с разными реализациями функций. В файле `lib.h` описаны контракты функций. В файлах `lab5-1.c` и `lab5-2.c` содержатся программы, выполняющие подключение библиотек на этапе компиляции и во время исполнения программы соответственно.

2 Исходный код

Файлы библиотек необходимо компилировать с ключами `-fPIC` и `-shared`, которые нужны для того, чтобы создать общий позиционно-независимый объект. Имя этого объекта должно иметь префикс `"lib"` и расширение `".so"`. Для подключения библиотеки на этапе компиляции необходимо явно указать компилятору на библиотеку. Для подключения во время исполнения необходимо использовать библиотеку `libdl`.

lib1.c

```
1  #include "lib.h"
2  #include <stdlib.h>
3
4  float E(int x){
5      float e=1;
6      for(int i=0; i<x; i++){
7          e*=1.0+1.0/(float)x;
8      }
9      return e;
10 }
11
12 char* Translation(long x){
13     int length=0, capacity=1, base = 2, diff = '0';
14     char* translation = (char*)malloc(2*sizeof(char));
15     translation[1] = '\0';
16     while(x>=base){
17         if(capacity==length){
18             capacity*=2;
19             translation=(char*)realloc(translation, (capacity+1)*sizeof(char));
20         }
21         translation[length]=x%base+'0';
22         x/=base;
23         length++;
24     }
25     if(capacity==length){
26         translation=(char*)realloc(translation, (length+2)*sizeof(char));
27     }
28     translation[length]=x+'0';
29     translation[length+1]='\0';
30     length++;
31     char tmp;
32     for(int i=0; i<length/2; i++){
33         tmp=translation[i];
34         translation[i]=translation[length-i-1];
35         translation[length-i-1]=tmp;
36     }
37     return translation;
38 }
```

lib2.c

```
1  //lib2
2
3  #include "lib.h"
4  #include <stdlib.h>
5
6  // -fPIC -shared
7
8  float E(int x){
9      unsigned long long fact=1;
10     float e=1.0;
11     for(int i=1; i<=x; i++){
```

```

12         e+=1.0/(float)fact;
13         fact*=fact+1;
14     }
15     return e;
16 }
17
18 char* Translation(long x){
19     int length=0, capacity=1, base = 3, diff = '0';
20     char* translation = (char*)malloc(2*sizeof(char));
21     translation[1] = '\0';
22     while(x>=base){
23         if(capacity==length){
24             capacity*=2;
25             translation=(char*)realloc(translation, (capacity+1)*sizeof(char));
26         }
27         translation[length]=x%base+'0';
28         x/=base;
29         length++;
30     }
31     if(capacity==length){
32         translation=(char*)realloc(translation, (length+2)*sizeof(char));
33     }
34     translation[length]=x+'0';
35     translation[length+1]='\0';
36     length++;
37     char tmp;
38     for(int i=0; i<length/2; i++){
39         tmp=translation[i];
40         translation[i]=translation[length-i-1];
41         translation[length-i-1]=tmp;
42     }
43     return translation;
44 }

```

lib.h

```

1 #ifndef LIB
2 #define LIB
3
4 float E(int x);
5
6 char* Translation(long x);
7
8 #endif
9 }

```

lab5-1.c

```

1 #include<stdlib.h>
2 #include<stdio.h>
3 #include"lib.h"
4
5 int main(){
6     int i;
7     printf("Enter function number and arguments:\n");
8     while(scanf("%d",&i)>0){
9         if(i==1){
10             while(scanf("%d",&i)>0){
11                 if(i>0){
12                     printf("%f\n", E(i));
13                 }
14                 else{
15                     break;

```

```

16         }
17     }
18     }else if(i==2){
19         char* string;
20         while(scanf("%d",&i)>0){
21             if(i>0){
22                 string=Translation(i);
23                 printf("%s\n",string);
24                 free(string);
25             }
26             else{
27                 break;
28             }
29         }
30     }
31 }
32 }

```

lab5-2.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<dlfcn.h>
4  #include"lib.h"
5
6  int main(int argc, char*argv[]){
7      char* firstlib=argv[1], *secondlib=argv[2];
8
9      void* lib1 = dlopen(firstlib, RTLD_LAZY);
10     void* lib2 = dlopen(secondlib, RTLD_LAZY);
11
12     int i=0, j=0;
13     if(lib1 == NULL){
14         perror("dlopen1");
15     }
16     if(lib2 == NULL){
17         perror("dlopen2");
18     }
19
20     float (*func1) (int);
21     char* (*func2) (long);
22     func1=dlsym(lib1, "E");
23     func2=dlsym(lib1, "Translation");
24     if(func1==NULL) perror("func1 error");
25     if(func2==NULL) perror("func2 error");
26
27     while(scanf("%d",&i)>0){
28         if(i==0){
29             if(j==0){
30                 printf("%s switched to %s\n", firstlib, secondlib);
31                 func1=dlsym(lib2, "E");
32                 func2=dlsym(lib2, "Translation");
33                 if(func1==NULL) perror("func1 error");
34                 if(func2==NULL) perror("func2 error");
35                 j++;
36             }
37             else{
38                 printf("%s switched to %s\n", secondlib, firstlib);
39                 func1=dlsym(lib1, "E");
40                 func2=dlsym(lib1, "Translation");
41                 if(func1==NULL) perror("func1 error");
42                 if(func2==NULL) perror("func2 error");
43                 j--;

```

```

44         }
45     }
46     else if(i==1){
47         while(scanf("%d",&i)>0){
48             printf("%f\n", func1(i));
49         }
50     }else if(i==2){
51         char* string;
52         while(scanf("%d",&i)>0){
53             string=func2(i);
54             printf("%s\n",string);
55             free(string);
56         }
57     }
58 }
59 dlclose(lib1);
60 dlclose(lib2);
61 }

```

3 Консоль

```
maloletniydebil@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OS/labs/lab5$ gcc lab5-1.c -lmylib1
maloletniydebil@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OS/labs/lab5$ ./a.out
Enter function number and arguments,
reading of arguments will be stopped when
you will enter something which willn't be integer
1
90
2.703333
900
2.716772
1000
2.716920
maloletniydebil@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OS/labs/lab5$ ./a.out
Enter function number and arguments,
reading of arguments will be stopped when
you will enter something which willn't be integer
2
5
101
10
1010
30
11110
maloletniydebil@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OS/labs/lab5$ gcc lab5-1.c -lmylib2
maloletniydebil@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OS/labs/lab5$ ./a.out
Enter function number and arguments:
1
90
2.691030
900
2.691030
89
2.691030
45
2.691030
20
2.691030
10
2.691030
7
2.691030
6
2.691030
5
2.691030
4
2.690476
```



```
3
2.666667
2
2.500000
1
2.000000
maloletniydebil@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OS/labs/lab5$ gcc lab5-2.c -ldl
maloletniydebil@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OS/labs/lab5$ ./a.out libmylib1.so libmylib2.so
1
90
2.703333
0
libmylib1.so switched to libmylib2.so
1
90
2.691030
2
2
2
0
libmylib2.so switched to libmylib1.so
2
2
10
```

4 Выводы

Выполнив лабораторную работу №5 по курсу «Операционные системы», я освоил работу с динамическими библиотеками в среде Unix (shared objects). Научился создавать библиотеки, подключать их во время компиляции на этапе линковки и во время выполнения программы.