Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу**

**«Операционные системы»**

**ДИНАМИЧЕСКИЕ БИБЛИОТЕКИ**

Студент: Почечура Артемий Андреевич

Группа: М8О–206Б–20

Вариант: 33

Преподаватель: Соколов Андрей Алексеевич

Оценка: 5

Дата: 4.12.21

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Постановка задачи**

**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 No1*), которая используют одну из библиотек, используя знания полученные на этапе компиляции;
* Тестовая программа (*программа No2*), которая загружает библиотеки, используя только их местоположение и контракты.

Провести анализ двух типов использования библиотек.

Пользовательский ввод для обоих программ должен быть организован следующим образом:

1. Если пользователь вводит команду «0», то программа переключает одну реализацию

контрактов на другую (необходимо только для *программы No2*). Можно реализовать лабораторную работу без данной функции, но максимальная оценка в этом случае будет «хорошо»;

1. «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;
2. «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 33.

6. Рассчет значения числа е(основание натурального логарифма Float E(int x) двумя способами: считая значение (1 + 1/x) ^ x, и считая сумму ряда по n от 0 до x, где элементы ряда равны: (1/(n!)).

9. Отсортировать целочисленный массив Int \* Sort(int \* array) двумя способами: сортировкой пузырьком и сортировкой Хоара.

В программе используются следующие системные вызовы:

1. **dlopen** – загружает динамическую библиотеку с указанным именем. В случае неуспеха возвращает NULL.
2. **dlclose –** уменьшает на единицу счетчик ссылок на указатель динамической библиотеки *handle*. Если нет других загруженных библиотек, использующих ее символы и если счетчик ссылок принимает нулевое значение, то динамическая библиотека выгружается.  В случае успеха возвращает 0, иначе ненулевой результат.
3. **dlsym** – использует указатель на динамическую библиотеку, возвращаемую dlopen, и оканчивающееся нулем символьное имя, а затем возвращает адрес, указывающий на нужный символ. В случае неуспеха dlsym возвращает NULL.
4. **dlerror** – возвращает сообщение об ошибке, если ошибки не произошло, то возвращает NULL.

**Основные файлы программы**

**Makefile:**

done: prog1 prog2

lib1.so: lib1.c

gcc -shared lib1.c -o lib1.so -Wall

lib2.so: lib2.c

gcc -shared lib2.c -o lib2.so -Wall

prog2: lib1.so lib2.so prog2.c

gcc prog2.c -ldl -o prog2 -Wall

prog1: lib1.so prog1.c

gcc prog1.c -L"/root" -Wl,-R. -l1 -l1 -o prog1 –Wall

**functions.h**

#ifndef FUNCTIONS\_H

#define FUNCTIONS\_H

float E(int x);

int\* Sort(int \*numbers, int left, int right);

#endif

**lib1.c**

#include <stdio.h>

#include <stdlib.h>

float E(int X){

float x=X;

float f=1+1/x;

float h=1;

while(x>0){

x--;

h=h\*f;

}

return h;

}

void Sort(int \*numbers, int r, int n){

for(int i=0;i<n;i++){

for(int j=0;j<n;j++){

if(numbers[j]>numbers[j+1]){

int y=numbers[j];

numbers[j]=numbers[j+1];

numbers[j+1]=y;

}

}

}

}

**lib2.c**

#include <stdio.h>

#include <stdlib.h>

float E(int x){

float f=1;

float k=1;

for(int i=1;i<=x;i++){

k=k\*i;

f=f+1/k;

}

return f;

}

void Sort(int \*numbers, int left, int right){

int pivot;

int l\_hold = left;

int r\_hold = right;

pivot = numbers[left];

while (left < right){

while ((numbers[right] >= pivot) && (left < right)){

right--;

}

if (left != right){

numbers[left] = numbers[right];

left++;

}

while ((numbers[left] <= pivot) && (left < right)){

left++;

}

if (left != right){

numbers[right] = numbers[left];

right--;

}

}

numbers[left] = pivot;

pivot = left;

left = l\_hold;

right = r\_hold;

if (left < pivot){

Sort(numbers, left, pivot - 1);

}

if (right > pivot){

Sort(numbers, pivot + 1, right);

}

}

**prog1.c**

#include <stdio.h>

#include "functions.h"

int main(){

printf("1 + float x - calculation of exponent with accuracy of x\n");

printf("2 + int n + int A[n] - number of members of array and array\n");

int command;

while(scanf("%d", &command) != EOF){

if(command==2){

int n;

if(scanf("%d", &n) != 1){

printf("Wrong arguements!\n");

continue;

}

int A[n];

for(int i=0;i<n;i++){

if(scanf("%d", &A[i]) != 1){

printf("Wrong arguements!\n");

continue;

}

}

Sort(A, 0, n-1);

for(int i=0;i<n;i++){

printf("%d ",A[i]);

}

printf("\n");

} else if(command==1){

float x;

if(scanf("%f", &x) != 1){

printf("Wrong arguements!\n");

continue;

}

printf("%f\n", E(x));

} else{

printf("Wrong command!\n");

}

}

}

**prog2.c**

#include <stdio.h>

#include <dlfcn.h>

int main(){

int version = 1;

int command;

float (\*E)(int) = NULL;

int (\*Sort)(int\*, int, int) = NULL;

void \*lib1\_handler = dlopen("./lib1.so",RTLD\_LAZY);

if (!lib1\_handler){

fprintf(stderr,"dlopen() error: %s\n", dlerror());

return -1;

}

void \*lib2\_handler = dlopen("./lib2.so",RTLD\_LAZY);

if (!lib2\_handler){

fprintf(stderr,"dlopen() error: %s\n", dlerror());

return -1;

}

E = dlsym(lib1\_handler,"E");

Sort = dlsym(lib1\_handler,"Sort");

printf("0 - to switch realizations\n");

printf("1 + float x - calculation of exponent with accuracy of x\n");

printf("2 + int n + int A[n] - number of members of array and array\n");

while(scanf("%d", &command) != EOF){

if(command==0){

if(version==2){

E = dlsym(lib1\_handler,"E");

Sort = dlsym(lib1\_handler,"Sort");

version=1;

} else{

E = dlsym(lib2\_handler,"E");

Sort = dlsym(lib2\_handler,"Sort");

version=2;

}

printf("Switched to realization %d\n", version);

} else if(command==1){

float x;

if(scanf("%f", &x) != 1){

printf("Wrong arguements!\n");

continue;

}

printf("%f\n", E(x));

} else if(command==2){

int n;

if(scanf("%d", &n) != 1){

printf("Wrong arguements!\n");

continue;

}

int A[n];

for(int i=0;i<n;i++){

if(scanf("%d", &A[i]) != 1){

printf("Wrong arguements!\n");

continue;

}

}

Sort(A, 0, n-1);

for(int i=0;i<n;i++){

printf("%d ",A[i]);

}

printf("\n");

} else{

printf("Wrong command!\n");

}

}

dlclose(lib1\_handler);

dlclose(lib2\_handler);

}

**Пример работы**

root@DESKTOP-5HM2HTK:~# ./prog1

1 + float x - calculation of exponent with accuracy of x

2 + int n + int A[n] - number of members of array and array

1 1000

2.717049

2

4

5 -3 0 20

-3 0 5 20

0

Wrong command!

root@DESKTOP-5HM2HTK:~# ./prog2

0 - to switch realizations

1 + float x - calculation of exponent with accuracy of x

2 + int n + int A[n] - number of members of array and array

1

1000

2.717049

0

Switched to realization 2

1

1000

2.718282

2

6

90 -5 20 -8 0 2

-8 -5 0 2 20 90

**Вывод**

В ходе выполнения данной лабораторной работы я познакомился с тем, как создавать и использовать динамические библиотеки. Стоить отметить, что в простых программах с минимальной функциональностью использование статических библиотек может быть предпочтительней, однако в больших программах, использующих несколько библиотек, использование динамических библиотек позволяет снизить потребление оперативной и дисковой памяти, поскольку динамическую библиотеку достаточно один раз выгрузить в память, что ею могли пользоваться все нуждающиеся программы.