**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Операционные системы»

**Лабораторная работа № 5**

Тема: Динамические библиотеки

Студент: Волков

Группа: 80-207

Преподаватель: Миронов Е. С.

Дата:

Оценка:

1. **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 используют 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 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 №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;

1. «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: 3, 9*

1. **Описание алгоритма**

Благодаря cmake, все заголовочные файлы компилируются сразу в динамические библиотеки. Эти библиотеки имею расширение .so. Запуская статическую, мы вводим вышеописанные команды и совершаются задачи, написанные в варианте. Запуская динамическую программу, ко всему этому еще добавляется команда смена вариант функционала: с медленного алгоритма решения задачи на быстрый.

1. **Наборы и результаты тестов**

*Тест 1: (статическая)*

1

3 123

Prime numbers: 29

2

Enter your array. SIZE = 10

>> 76 2 12 73 6 34 21 1 10 999

1 2 6 10 12 21 34 73 76 999

*Тест 2: (динамическая)*

> 0

Now im using lib2

> 1

Enter your a and b

>> 2 1000000

Prime nums: 78498

> 0

Now im using lib1

> 1

Enter your a and b

>> 2 1000000

Prime nums: 78498

Из этого теста можно увидеть, что вариант с быстрым поиском простых чисел работает за мгновение, а вариант с медленным - спустя 2 мин дал ответ. Это означает, что смена библиотеки сработала отлично.

*strace: (динамической)*

execve("./dynamic", ["./dynamic"], 0x7ffd90468020 /\* 19 vars \*/) = 0

brk(NULL) = 0x564b34eaf000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffeb8a7b180) = -1 EINVAL (Invalid argument)

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=36270, ...}) = 0

mmap(NULL, 36270, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f2f54fb5000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libdl.so.2", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 \22\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=18816, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f2f54fb3000

mmap(NULL, 20752, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f2f54fad000

mmap(0x7f2f54fae000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f2f54fae000

mmap(0x7f2f54fb0000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f2f54fb0000

mmap(0x7f2f54fb1000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f2f54fb1000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029224, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68

mmap(NULL, 2036952, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f2f54dbb000

mprotect(0x7f2f54de0000, 1847296, PROT\_NONE) = 0

mmap(0x7f2f54de0000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7f2f54de0000

mmap(0x7f2f54f58000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7f2f54f58000

mmap(0x7f2f54fa3000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f2f54fa3000

mmap(0x7f2f54fa9000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f2f54fa9000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f2f54db8000

arch\_prctl(ARCH\_SET\_FS, 0x7f2f54db8740) = 0

mprotect(0x7f2f54fa3000, 12288, PROT\_READ) = 0

mprotect(0x7f2f54fb1000, 4096, PROT\_READ) = 0

mprotect(0x564b346b2000, 4096, PROT\_READ) = 0

mprotect(0x7f2f54feb000, 4096, PROT\_READ) = 0

munmap(0x7f2f54fb5000, 36270) = 0

brk(NULL) = 0x564b34eaf000

brk(0x564b34ed0000) = 0x564b34ed0000

openat(AT\_FDCWD, "./liblib1.so", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`\20\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0777, st\_size=15912, ...}) = 0

getcwd("/mnt/d/Documents/Projects/c++/OC/lab5/cmake-wsl", 128) = 48

mmap(NULL, 16432, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f2f54fb9000

mmap(0x7f2f54fba000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f2f54fba000

mmap(0x7f2f54fbb000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f2f54fbb000

mmap(0x7f2f54fbc000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f2f54fbc000

close(3) = 0

mprotect(0x7f2f54fbc000, 4096, PROT\_READ) = 0

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x4), ...}) = 0

write(1, "> ", 2> ) = 2

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x4), ...}) = 0

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, 0x564b34eafc50, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

**read(0, 0**

**"0\n", 1024) = 2**

**munmap(0x7f2f54fb9000, 16432) = 0**

**write(1, "Now im using lib2\n", 18Now im using lib2**

**) = 18**

**openat(AT\_FDCWD, "./liblib2.so", O\_RDONLY|O\_CLOEXEC) = 3**

**read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\21\0\0\0\0\0\0"..., 832) = 832**

**fstat(3, {st\_mode=S\_IFREG|0777, st\_size=16512, ...}) = 0**

**getcwd("/mnt/d/Documents/Projects/c++/OC/lab5/cmake-wsl", 128) = 48**

**mmap(NULL, 16472, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f2f54fb9000**

**mmap(0x7f2f54fba000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f2f54fba000**

**mmap(0x7f2f54fbb000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f2f54fbb000**

**mmap(0x7f2f54fbc000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f2f54fbc000**

**close(3) = 0**

**mprotect(0x7f2f54fbc000, 4096, PROT\_READ) = 0**

**write(1, "> ", 2> ) = 2**

**read(0, 1**

**"1\n", 1024) = 2**

**write(1, "Enter your a and b\n", 19Enter your a and b**

**) = 19**

**write(1, ">> ", 3>> ) = 3**

**read(0, 2 10**

**"2 10\n", 1024) = 5**

**write(1, "Prime nums: 4\n", 14Prime nums: 4**

**) = 14**

**write(1, "> ", 2> ) = 2**

**read(0, 0**

**"0\n", 1024) = 2**

**munmap(0x7f2f54fb9000, 16472) = 0**

**write(1, "Now im using lib1\n", 18Now im using lib1**

**) = 18**

**openat(AT\_FDCWD, "./liblib1.so", O\_RDONLY|O\_CLOEXEC) = 3**

**read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`\20\0\0\0\0\0\0"..., 832) = 832**

**fstat(3, {st\_mode=S\_IFREG|0777, st\_size=15912, ...}) = 0**

**getcwd("/mnt/d/Documents/Projects/c++/OC/lab5/cmake-wsl", 128) = 48**

**mmap(NULL, 16432, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f2f54fb9000**

**mmap(0x7f2f54fba000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f2f54fba000**

**mmap(0x7f2f54fbb000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f2f54fbb000**

**mmap(0x7f2f54fbc000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f2f54fbc000**

**close(3) = 0**

**mprotect(0x7f2f54fbc000, 4096, PROT\_READ) = 0**

**write(1, "> ", 2> ) = 2**

**read(0, 1**

**"1\n", 1024) = 2**

**write(1, "Enter your a and b\n", 19Enter your a and b**

**) = 19**

**write(1, ">> ", 3>> ) = 3**

**read(0, 2 10**

**"2 10\n", 1024) = 5**

**write(1, "Prime nums: 4\n", 14Prime nums: 4**

**) = 14**

**write(1, "> ", 2> ) = 2**

**read(0, "", 1024) = 0**

**munmap(0x7f2f54fb9000, 16432) = 0**

**exit\_group(0) = ?**

+++ exited with 0 +++

1. **Листинг программы**

*child.c*

#include "functions.h"  
#include <stdio.h>  
#include <stdlib.h>  
  
int main(){  
 int command;  
 int a,b;  
 int arr[SIZE];  
 while(scanf("%d", &command) > 0){  
 if(command == 1){  
 if(scanf("%d %d", &a, &b) != 2){  
 perror("invalid input");  
 exit(1);  
 }  
 printf("Prime numbers: %d\n", PrimeCount(a,b));  
 fflush(stdout);  
 }  
  
 else if(command == 2){  
 printf("Enter your array. SIZE = %d\n>> ", SIZE);  
 fflush(stdout);  
 //array initialize  
 for (int i = 0; i < SIZE; ++i) {  
 scanf("%d", &arr[i]);  
 }  
  
 Sort(arr);  
 for (int i = 0; i < SIZE; ++i) {  
 printf("%d ", arr[i]);  
 }  
 printf("\n");  
 fflush(stdout);  
 }  
 }  
  
 return 0;  
}

*functions.h*

#ifndef LAB5\_FUNCTIONS\_H  
#define LAB5\_FUNCTIONS\_H  
  
#define SIZE 10  
extern int PrimeCount(int,int);  
  
extern void Sort(int arr[SIZE]);  
  
#endif //LAB5\_FUNCTIONS\_H

*function1.c*

#include "functions.h"  
  
void swap(int \*a, int \*b){  
 int tmp;  
 tmp = \*a;  
 \*a = \*b;  
 \*b = tmp;  
}  
  
void Sort(int arr[SIZE]) {  
 for (int i = 0; i < SIZE; ++i) {  
 for (int j = 0; j < SIZE - i - 1; ++j) {  
 if(arr[j] > arr[j+1] ){  
 swap(&arr[j],&arr[j+1]);  
 }  
 }  
 }  
}  
  
#define max(a, b) ((a) > (b)) ? (a) : (b)  
  
int PrimeCount(int A, int B) {  
 int n = 0, prime;  
 for (int num = max(A, 2); num <= B; ++num) {  
 prime = 1;  
 for (int i = 2; i < num; ++i) {  
 if (num % i == 0) {  
 prime = 0;  
 break;  
 }  
 }  
 if (prime == 1) {  
 ++n;  
 }  
 }  
 return n;  
}

*function2.c*

#include "functions.h"  
#include "stdlib.h"  
#include "stdio.h"  
  
void swap(int \*a, int \*b){  
 int tmp;  
 tmp = \*a;  
 \*a = \*b;  
 \*b = tmp;  
}  
  
void Qsort(int arr[SIZE], int lhs, int rhs){  
 int l = lhs;  
 int r = rhs;  
 int rep = arr[(l + r)/2];  
  
 while(l <= r){  
 while(arr[l] < rep){  
 ++l;  
 }  
 while(arr[r] > rep){  
 --r;  
 }  
 if(l <= r){  
 swap(&arr[l++], &arr[r--]);  
 }  
 }  
  
 if(lhs < r){  
 Qsort(arr, lhs, r);  
 }  
 if(l < rhs){  
 Qsort(arr,l,rhs);  
 }  
}  
  
void Sort(int arr[SIZE]){  
 Qsort(arr,0,SIZE - 1);  
}  
  
int PrimeCount(int A, int B) {  
 int n = 0;  
 int \*sieve = (int \*)malloc(sizeof(int) \* (B+1));  
 if (sieve == NULL) {  
 perror("malloc error");  
 exit(1);  
 }  
 for (int i = 0; i <= B; ++i) sieve[i] = 0;  
 for (int i = 2; i <= B; ++i) {  
 if (sieve[i] != 0) {  
 continue;  
 }  
 ++n;  
 for (int j = i + i; j <= B; j += i) {  
 sieve[j] = 1;  
 }  
 }  
 return n;  
}

*dynamic.c*

#include "functions.h"  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <dlfcn.h>  
  
#define check(VALUE, MSG, BADVAL) if (VALUE == BADVAL) { perror(MSG); exit(1); }  
  
int main(){  
 char\* libs[] = {"./liblib1.so", "./liblib2.so"};  
 int lib = 0;  
 int (\*PrimeCount)(int, int) = NULL;  
 void (\*Sort)(int [SIZE]) = NULL;  
  
 void\* handle = dlopen(libs[lib], RTLD\_NOW);  
 check(handle,dlerror(),NULL);  
  
 int command;  
 int a,b;  
 int arr[SIZE];  
 while((printf("> ") && fflush(stdout)) || scanf("%d", &command) > 0){  
 if(command == 0){  
 if (dlclose(handle) != 0) {  
 perror(dlerror());  
 exit(1);  
 }  
 lib = (lib + 1) % 2;  
 if(lib == 0){  
 printf("Now im using lib1\n");  
 fflush(stdout);  
 }  
 else{  
 printf("Now im using lib2\n");  
 fflush(stdout);  
 }  
 handle = dlopen(libs[lib], RTLD\_NOW);  
 check(handle, dlerror(), NULL);  
 }  
  
 else if(command == 1){  
 printf("Enter your a and b\n>> ");  
 fflush(stdout);  
 if(scanf("%d %d", &a, &b) != 2){  
 perror("invalid arguments");  
 exit(1);  
 }  
  
 PrimeCount = dlsym(handle, "PrimeCount");  
 check(PrimeCount,dlerror(),NULL);  
 printf("Prime nums: %d\n", (\*PrimeCount)(a,b));  
 fflush(stdout);  
 }  
 else if(command == 2){  
 printf("Enter your array. SIZE = %d\n>> ", SIZE);  
 fflush(stdout);  
 //array initialize  
 for (int i = 0; i < SIZE; ++i) {  
 scanf("%d", &arr[i]);  
 }  
  
 Sort = dlsym(handle, "Sort");  
 check(Sort, dlerror(), NULL);  
 (\*Sort)(arr);  
  
 printf("array is:\n");  
 fflush(stdout);  
 for (int i = 0; i < SIZE; ++i) {  
 printf("%d ", arr[i]);  
 }  
 printf("\n");  
 fflush(stdout);  
 }  
 }  
  
 if (dlclose(handle) != 0) {  
 perror(dlerror());  
 exit(1);  
 }  
  
 return 0;  
}

1. **Вывод**

Благодаря проделанной работе я на практике узнал что такое динамические библиотеки, как они создаются и зачем они нужны. Во время работы, я также узнал, что .dll в windows является расширение для динамической библиотеки и понял зачем нужно так много .dll файлов в папках с играми.

**Список литературы**

1. *Анатомия динамических библиотек Linux — IBM*

URL: <https://www.ibm.com/developerworks/ru/library/l-dynamic-libraries/>

(дата обращения 22.12.2020)

*2. Shared libraries with GCC on Linux — Cprogramming.com*

URL: <https://www.cprogramming.com/tutorial/shared-libraries-linux-gcc.html>

(дата обращения 22.12.2020)

*3. dlopen(3) — Linux manual page — man7.org*

URL: <https://www.man7.org/linux/man-pages/man3/dlopen.3.html>

(дата обращения 22.12.2020)

*4. dlsym(3) — Linux manual page — man7.org*

URL: <https://man7.org/linux/man-pages/man3/dlsym.3.html>

(дата обращения 22.12.2020)