

Московский Авиационный Институт  
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики  
Кафедра вычислительной математики и программирования

**Лабораторная работа №5**  
**по курсу «Операционные системы»**  
**III Семестр**

**Вариант 15**

Студент:	Короткевич Л. В.
Группа:	М80-208Б-19
Преподаватель:	Миронов Е.С
Оценка:	
Дата:	

# 1. Постановка задачи

**Тема:** Динамические библиотеки

**Цель работы** - приобретение практических навыков в:

- Создании динамических библиотек
- Создании 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.

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 15:** контракты 2 - расчет производной функции  $\cos(x)$ , 9 – сортировка целочисленного массива.

## 2. Метод решения

Используемые системные вызовы для выполнения работы:

<b>void *dlopen(const char *filename, int flag);</b>	dlopen загружает динамическую библиотеку, имя которой указано в строке filename, и возвращает прямой указатель на начало динамической библиотеки.
<b>int dlclose(void *handle);</b>	dlclose уменьшает на единицу счетчик ссылок на указатель динамической библиотеки handle.
<b>void *dlsym(void *handle, char *symbol);</b>	dlsym использует указатель на динамическую библиотеку, возвращаемую dlopen, и оканчивающееся нулем символьное имя, а затем возвращает адрес, указывающий, откуда загружается этот символ.

Создание и применение shared library на примере Solyanka1:

1. Необходимо описать функции в Solyanka1.c, прототипы в Solyanka1.h

2. Далее – получить объектник Solyanka1.o:

```
gcc -fPIC -c Solyanka1.c -o Solyanka1.o
```

\* *-fPIC – Position Independent Code*

\* *-g для добавления дебаг-информации*

\* *-c — чтобы создать только файл библиотеки, без вызова линкера*

3. Теперь создаем общую библиотеку Solyanka1.so:

```
gcc -shared Solyanka1.o -o libSolyanka1.so
```

\* *-shared – тип библиотеки – общая*

\* *-o – сохранить результат в Solyanka1.so*

4. Собираем программу:

```
gcc main1.c -L. -lSolyanka1 -o main1
```

\* *-lSolyanka1 – указываем название библиотеки*

\* *-L. — путь, по которому необходимо искать файл библиотеки – ./*

5. Добавляем текущий каталог в \$LD\_LIBRARY\_PATH:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
```

## 3. Тестирование

```
[leo@pc final]$ make
```

```
gcc -fPIC -c Solyanka1.c -o Solyanka1.o
```

```
gcc -shared Solyanka1.o -o libSolyanka1.so
```

```
gcc main1.c -L. -lSolyanka1 -o main1
```

```
gcc -fPIC -c Solyanka2.c -o Solyanka2.o
```

```
gcc -shared Solyanka2.o -o libSolyanka2.so
```

```
gcc main2.c -ldl -o main2
```

```
[leo@pc final]$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
```

```
[leo@pc final]$ ./main1
```

```
Enter function type: 1
```

```
Enter x: 3.1415
```

```
Enter dx: 0.001
```

```
0.000536
```

```
Enter function type: 2
```

```

Enter number of elements: 10
Enter 10 elements to sort: 0 1 3 5 2 1 3 4 5 5
Sorted array: 0 1 1 2 3 3 4 5 5 5 Enter function type: [leo@pc final]$
[leo@pc final]$ ./main2
Enter function type: 0
Implementations switched. Enter function type again: 1
Enter x: 3.14
Enter dx: 0.01
-0.001591
Enter function type: 0
Implementations switched. Enter function type again: 1
Enter x: 3.14
Enter dx: 0.01
0.003392
Enter function type: 2
Enter number of elements: 3
Enter 3 elements to sort: 3 1 2
Sorted array: 1 2 3 Enter function type: 0
Implementations switched. Enter function type again: 2
Enter number of elements: 5
Enter 5 elements to sort: 0 1 4 2 -3
Sorted array: -3 0 1 2 4

```

## **strace, ltrace:**

### **Отображение всех вызовов – Strace:**

```

[leo@pc final]$ strace -f ./main2
execve("./main2", ["/main2"], 0x7ffca8a66f8 /* 70 vars */) = 0
brk(NULL)                               = 0x556e3492f000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd75fb18b0) = -1 EINVAL (Invalid
argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/haswell/x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/haswell/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD, "haswell/x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "haswell/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "libdl.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)

```

```

openat(AT_FDCWD, "./tls/haswell/x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "./tls/haswell/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./tls/x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./tls/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD, "./haswell/x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
openat(AT_FDCWD, "./haswell/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./x86_64/libdl.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./libdl.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 6
fstat(6, {st_mode=S_IFREG|0644, st_size=221255, ...}) = 0
mmap(NULL, 221255, PROT_READ, MAP_PRIVATE, 6, 0) = 0x7f555515f000
close(6) = 0
openat(AT_FDCWD, "/usr/lib/libdl.so.2", O_RDONLY|O_CLOEXEC) = 6
read(6, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20\22\0\0\0\0\0"..., 832) =
832
fstat(6, {st_mode=S_IFREG|0755, st_size=18608, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_ANONYMOUS, -1, 0) = 0x7f555515d000
mmap(NULL, 20624, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 6, 0) =
0x7f5555157000
mmap(0x7f5555158000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x1000) = 0x7f5555158000
mmap(0x7f555515a000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 6, 0x3000) = 0x7f555515a000
mmap(0x7f555515b000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x3000) = 0x7f555515b000
close(6) = 0
openat(AT_FDCWD, "tls/haswell/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/haswell/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "haswell/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)

```

```

openat(AT_FDCWD, "haswell/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "./tls/haswell/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
openat(AT_FDCWD, "./tls/haswell/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD, "./haswell/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./haswell/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "./libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 6
read(6, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\202\2\0\0\0\0\0"..., 832)
= 832
pread64(6, "\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(6, "\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(6, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\207\360\21\247\344\314?\306\nT\
320\323\335i\16t"..., 68, 880) = 68
fstat(6, {st_mode=S_IFREG|0755, st_size=2159552, ...}) = 0
pread64(6, "\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
mmap(NULL, 1868448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 6, 0)
= 0x7f5554f8e000
mmap(0x7f5554fb4000, 1363968, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x26000) = 0x7f5554fb4000
mmap(0x7f5555101000, 311296, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 6, 0x173000) = 0x7f5555101000
mmap(0x7f555514d000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x1be000) = 0x7f555514d000
mmap(0x7f5555153000, 12960, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5555153000
close(6)
= 0

```

```

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_ANONYMOUS, -1, 0) = 0x7f5554f8b000
arch_prctl(ARCH_SET_FS, 0x7f5554f8b740) = 0
mprotect(0x7f555514d000, 12288, PROT_READ) = 0
mprotect(0x7f555515b000, 4096, PROT_READ) = 0
mprotect(0x556e34784000, 4096, PROT_READ) = 0
mprotect(0x7f55551c2000, 4096, PROT_READ) = 0
munmap(0x7f555515f000, 221255) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x2), ...}) = 0
brk(NULL) = 0x556e3492f000
brk(0x556e34950000) = 0x556e34950000
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x2), ...}) = 0
write(1, "Enter function type: ", 21Enter function type: ) = 21
read(0, 1
"1\n", 1024) = 2
openat(AT_FDCWD, "tls/haswell/x86_64/libSolyanka1.so", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/haswell/libSolyanka1.so", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/x86_64/libSolyanka1.so", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/libSolyanka1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "haswell/x86_64/libSolyanka1.so", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "haswell/libSolyanka1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "x86_64/libSolyanka1.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "libSolyanka1.so", O_RDONLY|O_CLOEXEC) = 6
read(6, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\20\0\0\0\0\0"..., 832) =
832
fstat(6, {st_mode=S_IFREG|0755, st_size=16176, ...}) = 0
getcwd("/home/leo/programming/University/OS/Lab5/final", 128) = 47
mmap(NULL, 16472, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 6, 0) =
0x7f5555191000
mmap(0x7f5555192000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x1000) = 0x7f5555192000
mmap(0x7f5555193000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 6, 0x2000) = 0x7f5555193000
mmap(0x7f5555194000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x2000) = 0x7f5555194000
close(6) = 0
mprotect(0x7f5555194000, 4096, PROT_READ) = 0
write(1, "Enter x: ", 9Enter x: ) = 9

```

```

read(0, 3.14
"3.14\n", 1024)          = 5
write(1, "Enter dx: ", 10Enter dx: )          = 10
read(0, 0.01
"0.01\n", 1024)          = 5
write(1, "0.003392\n", 90.003392
)          = 9
write(1, "Enter function type: ", 21Enter function type: ) = 21
read(0, 0
"0\n", 1024)              = 2
munmap(0x7f5555191000, 16472)          = 0
write(1, "Implementations switched. Enter "..., 53Implementations switched. Enter
function type again: ) = 53
read(0, 1
"1\n", 1024)              = 2
openat(AT_FDCWD, "tls/haswell/x86_64/libSolyanka2.so", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/haswell/libSolyanka2.so", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/x86_64/libSolyanka2.so", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
openat(AT_FDCWD, "tls/libSolyanka2.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "haswell/x86_64/libSolyanka2.so", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "haswell/libSolyanka2.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "x86_64/libSolyanka2.so", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD, "libSolyanka2.so", O_RDONLY|O_CLOEXEC) = 6
read(6, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200\20\0\0\0\0\0"..., 832) =
832
fstat(6, {st_mode=S_IFREG|0755, st_size=16096, ...}) = 0
getcwd("/home/leo/programming/University/OS/Lab5/final", 128) = 47
mmap(NULL, 16464, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 6, 0) =
0x7f5555191000
mmap(0x7f5555192000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x1000) = 0x7f5555192000
mmap(0x7f5555193000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 6, 0x2000) = 0x7f5555193000
mmap(0x7f5555194000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 6, 0x2000) = 0x7f5555194000
close(6)          = 0
mprotect(0x7f5555194000, 4096, PROT_READ) = 0
write(1, "Enter x: ", 9Enter x: )          = 9

```



```

read(0, 3.14
"3.14\n", 1024)          = 5
write(1, "Enter dx: ", 10Enter dx: )          = 10
read(0, 0.01
"0.01\n", 1024)          = 5
write(1, "-0.001591\n", 10-0.001591
)          = 10
write(1, "Enter function type: ", 21Enter function type: ) = 21
read(0, "", 1024)          = 0
munmap(0x7f5555191000, 16464)          = 0
exit_group(0)              = ?
+++ exited with 0 +++

```

### Отображение всех вызовов -- ltrace:

```

[leo@pc final]$ ltrace ./main1
printf("Enter function type: ")          = 21
__isoc99_scanf(0x55913d3ed043, 0x7ffc99264b40, 0, 0Enter function type: 1
)          = 1
printf("Enter x: ")          = 9
__isoc99_scanf(0x55913d3ed016, 0x7ffc99264b44, 0, 0Enter x: 3.14
)          = 1
printf("Enter dx: ")          = 10
__isoc99_scanf(0x55913d3ed016, 0x7ffc99264b48, 0, 0Enter dx: 0.01
)          = 1
printf("%f\n", 0.0033920.003392
)          = 9
printf("Enter function type: ")          = 21
__isoc99_scanf(0x55913d3ed043, 0x7ffc99264b40, 0, 0Enter function type: )
= 0xffffffff
+++ exited (status 0) +++

```

---

```

[leo@pc final]$ ltrace ./main2
printf("Enter function type: ")          = 21
__isoc99_scanf(0x55cd6c98803e, 0x7ffefff95e14, 0, 0Enter function type: 1
)          = 1
dlopen("libSolyanka1.so", 1)          =
0x55cd6d75aae0
dlsym(0x55cd6d75aae0, "sort")          =
0x7f8c52484449
dlsym(0x55cd6d75aae0, "der")          =
0x7f8c52484287
printf("Enter x: ")          = 9
__isoc99_scanf(0x55cd6c988088, 0x7ffefff95e18, 0, 0Enter x: 3.14
)          = 1
printf("Enter dx: ")          = 10
__isoc99_scanf(0x55cd6c988088, 0x7ffefff95e1c, 0, 0Enter dx: 0.01
)          = 1
printf("%f\n", 0.0033920.003392
)          = 9
printf("Enter function type: ")          = 21
__isoc99_scanf(0x55cd6c98803e, 0x7ffefff95e14, 0, 0Enter function type: 0

```

```

)                                = 1
dlclose(0x55cd6d75aae0)                                = 0
printf("Implementations switched. Enter "...           = 53
__isoc99_scanf(0x55cd6c98803e, 0x7ffefff95e14, 0, 0Implementations switched. Enter function
type again: 1
)                                = 1
dlopen("libSolyanka2.so", 1)                            =
0x55cd6d75aae0
dlsym(0x55cd6d75aae0, "sort")                            =
0x7f8c524843cc
dlsym(0x55cd6d75aae0, "der")                            =
0x7f8c52484277
printf("Enter x: ")                                    = 9
__isoc99_scanf(0x55cd6c988088, 0x7ffefff95e18, 0, 0Enter x: 3.14
)                                = 1
printf("Enter dx: ")                                    = 10
__isoc99_scanf(0x55cd6c988088, 0x7ffefff95e1c, 0, 0Enter dx: 0.01
)                                = 1
printf("%f\n", -0.001591-0.001591
)                                = 10
printf("Enter function type: ")                        = 21
__isoc99_scanf(0x55cd6c98803e, 0x7ffefff95e14, 0, 0Enter function type: 1
)                                = 1
dlopen("libSolyanka2.so", 1)                            =
0x55cd6d75aae0
dlsym(0x55cd6d75aae0, "sort")                            =
0x7f8c524843cc
dlsym(0x55cd6d75aae0, "der")                            =
0x7f8c52484277
printf("Enter x: ")

```

### Статистика системных вызовов – strace:

```
[leo@pc final]$ strace -c ./main2
```

```
Enter function type: 0
```

```
Implementations switched. Enter function type again: 1
```

```
Enter x: 3.14
```

```
Enter dx: 0.01
```

```
-0.001591
```

```
Enter function type: 0
```

```
Implementations switched. Enter function type again: 1
```

```
Enter x: 3.14
```

```
Enter dx: 0.01
```

```
0.003392
```

```
Enter function type: % time    seconds  usecs/call    calls    errors syscall
```

```

-----
26.65  0.000504    9    51    46 openat
23.22  0.000439   33    13     read
15.34  0.000290   14    20    mmap
11.85  0.000224   20    11    write
 6.29  0.000119   19     6  mprotect
 5.92  0.000112   37     3  munmap
 3.12  0.000059    8     7   fstat
 3.07  0.000058   14     4  pread64

```

1.64	0.000031	10	3	brk
1.48	0.000028	5	5	close
0.74	0.000014	7	2	getcwd
0.69	0.000013	6	2	1 arch_prctl
0.00	0.000000	0	1	1 access
0.00	0.000000	0	1	execve

---

100.00	0.001891	14	129	48 total
--------	----------	----	-----	----------

### Статистика системных вызовов - ltrace:

[leo@pc final]\$ ltrace -c ./main2

Enter function type: 0

Implementations switched. Enter function type again: 1

Enter x: 3.14

Enter dx: 0.01

-0.001591

Enter function type: 0

Implementations switched. Enter function type again: 1

Enter x: 3.14

Enter dx: 0.01

0.003392

Enter function type:	% time	seconds	usecs/call	calls	function
----------------------	--------	---------	------------	-------	----------

---

99.83	13.430960	1492328	9	__isoc99_scanf
0.13	0.016937	8468	2	dlopen
0.03	0.003364	305	11	printf
0.01	0.001876	938	2	dlclose
0.00	0.000331	82	4	dlsym

---

100.00	13.453468	28 total
--------	-----------	----------

## 4. Листинг программы

### main1.c:

```
#include <stdio.h>
```

```
#include "Solyanka1.c"
```

```
int main()
```

```
{
```

```
    int type;
```

```
    while (printf("Enter function type: ") && scanf("%d", &type) > 0)
```

```
    {
```

```
        if (type == 1)
```

```
        {
```

```
            // cos'(x)
```

```
            float res, x, dx;
```

```
            printf("Enter x: ");
```

```
            scanf("%f", &x);
```

```
            printf("Enter dx: ");
```

```
            scanf("%f", &dx);
```

```
            float ans = der(x, dx);
```

```
            printf("%f\n", ans);
```

```
        }
```

```
        if (type == 2)
```

```
        {
```

```
            // qsort
```

```
            int n;
```

```
            printf("Enter number of elements: ");
```

```
            scanf("%d", &n);
```

```
            int *arr = (int *)malloc(sizeof(int) * n);
```

```
            printf("Enter %d elements to sort: ", n);
```

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

```

        scanf("%d", &arr[i]);

    int *res = sort(arr, n);
    printf("Sorted array: ");
    for (int i = 0; i < n; ++i)
        printf("%d ", res[i]);
    printf("\n");
}

return 0;
}

```

## main2.c:

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdint.h>
#include <dlfcn.h>

#include "Solyanka.h"

int main()
{
    void *library_handler = NULL;
    int type, impl = 0;
    while (printf("Enter function type: ") && scanf("%d", &type) > 0)
    {
        if (type == 0)
        {
            // switch implementations
            if (library_handler)
                dlclose(library_handler);
            impl = 1 - impl;
            printf("Implementations switched. Enter function type again: ");
            scanf("%d", &type);
        }

        if (impl == 0)
        {
            library_handler = dlopen("libSolyanka1.so", RTLD_LAZY);
            if (!library_handler)
            {
                fprintf(stderr, "dlopen() error: %s\n", dlerror());
                exit(1);
            }
        }
        else if (impl == 1)
        {
            library_handler = dlopen("libSolyanka2.so", RTLD_LAZY);
            if (!library_handler)
            {
                fprintf(stderr, "dlopen() error: %s\n", dlerror());
                exit(1);
            }
        }
    }

    int (*sort)(int *arr, int n);
    float (*der)(float x, float dx);

    sort = dlsym(library_handler, "sort");
    der = dlsym(library_handler, "der");

    if (type == 1)
    {
        // cos'(x)
        float res, x, dx;
        printf("Enter x: ");
        scanf("%f", &x);
        printf("Enter dx: ");
        scanf("%f", &dx);
        float ans = (*der)(x, dx);
        printf("%f\n", ans);
    }
    else if (type == 2)
    {
        // qsort
        int n;
        printf("Enter number of elements: ");
    }
}

```

```

scanf("%d", &n);

int *arr = (int *)malloc(sizeof(int) * n);
printf("Enter %d elements to sort: ", n);
for (int i = 0; i < n; ++i)
    scanf("%d", &arr[i]);

int *res = (*sort)(arr, n);
printf("Sorted array: ");
for (int i = 0; i < n; ++i)
    printf("%d ", res[i]);
printf("\n");
}
}

dlclose(library_handler);

return 0;
}

```

### **Solyanka.h:**

#pragma once

```

int *sort(int *array, int n);
float der(float x, float dx);

```

### **Solyanka1.c:**

```

#include "Solyanka.h"
#include <stdlib.h>
#include <string.h>

```

// derivative

```
const float pi = 3.1415926535897;
```

```
float cosine(float x)
```

```

{
    int div = (int)(x / pi);
    x = x - (div * pi);
    char sign = 1;
    if (div % 2 != 0)
        sign = -1;

```

```

    float result = 1.0;
    float inter = 1.0;
    float num = x * x;
    for (int i = 1; i <= 16; i++)
    {
        float comp = 2.0 * i;
        float den = comp * (comp - 1.0);
        inter *= num / den;
        if (i % 2 == 0)
            result += inter;
        else
            result -= inter;
    }

```

```

    return sign * result;
}

```

```
float der(float x, float dx)
```

```
{
    float res;
    res = (cosine((x + dx)) - cosine(x)) / dx;
    return res;
}
```

// sorting

```
void swap(int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}
```

```
int partition(int *arr, int low, int high)
{
    int pivot = arr[high];
    int i = (low - 1);

    for (int j = low; j <= high - 1; j++)
    {
        if (arr[j] < pivot)
        {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);

    return (i + 1);
}
```

```
void qs(int *arr, int low, int high)
{
    if (low < high)
    {
        int pi = partition(arr, low, high);
        qs(arr, low, pi - 1);
        qs(arr, pi + 1, high);
    }
}
```

```
int *sort(int *arr, int n)
{
    int *res = (int *)malloc(sizeof(int) * n);
    memcpy(res, arr, n * sizeof(int));
    qs(res, 0, n - 1);
    return res;
}
```

**Solyanka2.c:**

```
#include "Solyanka.h"
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
// derivative
```

```
const float pi = 3.1415926535897;
```

```
float cosine(float x)
```

```
{
```

```
    int div = (int)(x / pi);
```

```
    x = x - (div * pi);
```

```
    char sign = 1;
```

```
    if (div % 2 != 0)
```

```
        sign = -1;
```

```
    float result = 1.0;
```

```
    float inter = 1.0;
```

```
    float num = x * x;
```

```
    for (int i = 1; i <= 16; i++)
```

```
    {
```

```
        float comp = 2.0 * i;
```

```
        float den = comp * (comp - 1.0);
```

```
        inter *= num / den;
```

```
        if (i % 2 == 0)
```

```
            result += inter;
```

```
        else
```

```
            result -= inter;
```

```
    }
```

```
    return sign * result;
```

```
}
```

```
float der(float x, float dx)
```

```
{
```

```
    float res;
```

```
    res = (cosine((x + dx)) - cosine((x - dx))) / (2 * dx);
```

```
    return res;
```

```
}
```

```
// sorting
```

```
void swap(int *a, int *b)
```

```
{
```

```
    int t = *a;
```

```
    *a = *b;
```

```
    *b = t;
```

```
}
```

```
void bs(int *arr, int n)
```

```
{
```

```

int i, j;
for (i = 0; i < n - 1; i++)
    for (j = 0; j < n - i - 1; j++)
        if (arr[j] > arr[j + 1])
            swap(&arr[j], &arr[j + 1]);
}

int *sort(int *arr, int n)
{
    int *res = (int *)malloc(sizeof(int) * n);
    memcpy(res, arr, n * sizeof(int));
    bs(res, n);
    return res;
}

```

### **Makefile:**

```
all: main1 main2
```

```
main1: libSolyanka1.so main1.c
    gcc main1.c -L. -lSolyanka1 -o main1
```

```
main2: libSolyanka1.so libSolyanka2.so main2.c
    gcc main2.c -ldl -o main2
```

```
libSolyanka1.so: Solyanka1.o
    gcc -shared Solyanka1.o -o libSolyanka1.so
```

```
libSolyanka2.so: Solyanka2.o
    gcc -shared Solyanka2.o -o libSolyanka2.so
```

```
Solyanka1.o:
    gcc -fPIC -c Solyanka1.c -o Solyanka1.o
```

```
Solyanka2.o:
    gcc -fPIC -c Solyanka2.c -o Solyanka2.o
```

```
clean:
```

```

rm -f Solyanka1.o
rm -f libSolyanka1.so
rm -f Solyanka2.o
rm -f libSolyanka2.so
rm -f main1
rm -f main2

```



## Вывод

По мере выполнения данной лабораторной работы я описал динамические библиотеки функций сортировки целочисленного массива, подсчета значения производной функции  $\cos(x)$  в двух реализациях. Описал программу, которая использует функции динамических библиотек.

Использовал эти динамические библиотеки двумя способами: во время компиляции (на этапе линковки) и во время исполнения программы.

Описал также переключение реализации контрактов на другую во время работы программы (той, которая загружает библиотеки, используя только их местоположение и контракты).

По мере изучения материала на данную тему, я выяснил, что динамическая библиотека — это по сути исполняемый файл. Процесс динамической линковки запускается в момент создания процесса. И, как не трудно догадаться, в таком случае, исполняемые файлы линкуются между собой каждый раз, когда создается новый процесс — этот случай соответствует первой реализации программы моей ЛР.

Но также возможен вариант дин. линковки после запуска нового процесса, когда некоторая библиотека подгружается в адресное пространство уже работающего процесса — этот случай соответствует второй реализации программы моей ЛР.

Статическая линковка же, по сути, линкует объектные с конкретным файлом. Из-за чего возрастает размер условного инсталлятора; но зато этот метод надёжен.