

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

**Лабораторная работа №8 по курсу
«Операционные системы»**

Студент: Юнусов Руслан Асифович
Группа: М8О-209Б-23
Вариант: 24
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2024

Содержание

Репозиторий.....	3
Постановка задачи.....	3
Выполнение задания.....	3
Выводы.....	18

Репозиторий

<https://github.com/Rissochek/OSLabs/tree/main/lab8>

Постановка задачи

Цель работы

Приобретение практических навыков диагностики работы программного обеспечения.

Задание

При выполнении лабораторных работ по курсу ОС необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР. По итогам выполнения всех лабораторных работ отчет по данной ЛР должен содержать краткую сводку по исследованию написанных программ.

Выполнение задания

Lab1

```
rissochek@admin:/mnt/c/Users/rusla/coding/OSLabs/lab1$ strace ./parent test.txt
execve("./parent", [".parent", "test.txt"], 0x7fff62bb9ab8 /* 35 vars */) = 0
brk(NULL)                               = 0x55f9a45ca000
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbb9c6000
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=22943, ...}) = 0
mmap(NULL, 22943, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fbb9c0000
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\3\0>\0\1\0\0\0\220\243\2\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"...,
784, 64) = 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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"...,
784, 64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7fbb9c7ae000
```

```

mmap(0x7fbb7d6000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7fbb7d6000
mmap(0x7fbb95e000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) =
0x7fbb95e000
mmap(0x7fbb9ad000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) =
0x7fbb9ad000
mmap(0x7fbb9b3000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fbb9b3000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbb7ab000
arch_prctl(ARCH_SET_FS, 0x7fbb7ab740) = 0
set_tid_address(0x7fbb7aba10) = 4026
set_robust_list(0x7fbb7aba20, 24) = 0
rseq(0x7fbb7ac060, 0x20, 0, 0x53053053) = 0
mprotect(0x7fbb9ad000, 16384, PROT_READ) = 0
mprotect(0x55f974d44000, 4096, PROT_READ) = 0
mprotect(0x7fbb9fe000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7fbb9c0000, 22943) = 0
getrandom("\x8b\x96\x63\xaf\x4c\x14\xe5\x8e", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x55f9a45ca000
brk(0x55f9a45eb000) = 0x55f9a45eb000
pipe2([3, 4], 0) = 0
pipe2([5, 6], 0) = 0
clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fbb7aba10) = 4027
close(3) = 0
close(6) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x4), ...}) = 0
read(0, 2 2
"2 2\n", 1024) = 4
write(4, "test.txt", 8) = 8
write(4, "|", 1) = 1
write(4, "2 2", 3) = 3
close(4) = 0
close(5) = 0

```

```
wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 4027
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=4027,
si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---
exit_group(0) = ?
+++ exited with 0 +++
```

Системные вызовы

1. **fork** - создает копию текущего процесса, который является дочерним процессом для текущего процесса
2. **pipe** - создаёт однонаправленный канал данных, который можно использовать для взаимодействия между процессами.
3. **dup2** - перенаправляет вывод родительского файла в дочерний файл, а также вывод дочернего файла в родительский
4. **execv** - запускает дочерний процесс из отдельного файла.
5. **close** - закрывает файл, а также файловые дескрипторы.
6. **read** - читает количество байт(третий аргумент) из файла с файловым дескриптором(первый аргумент) в область памяти(второй аргумент).
7. **write** - записывает в файл с файловым дескриптором(первый аргумент) из области памяти(второй аргумент) количество байт(третий аргумент).
8. **perror** – вывод сообщения об ошибке.
9. **exit** - завершает выполнение программы.
10. **wait** - получает статус завершения дочернего процесса.

Lab2

```
rissochek@admin:/mnt/c/Users/rusla/coding/OSLabs/lab2$ strace ./a.out 1
execve("./a.out", ["/a.out", "1"], 0x7ffe2e327318 /* 35 vars */) = 0
brk(NULL) = 0x55fa800fd000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f52cbe93000
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=22943, ...}) = 0
mmap(NULL, 22943, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f52cbe8d000
```

```

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\3\0>\0\1\0\0\0\220\243\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
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7f52cbc7b000
mmap(0x7f52cbca3000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7f52cbca3000
mmap(0x7f52cbe2b000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) =
0x7f52cbe2b000
mmap(0x7f52cbe7a000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) =
0x7f52cbe7a000
mmap(0x7f52cbe80000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f52cbe80000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f52cbc78000
arch_prctl(ARCH_SET_FS, 0x7f52cbc78740) = 0
set_tid_address(0x7f52cbc78a10) = 9086
set_robust_list(0x7f52cbc78a20, 24) = 0
rseq(0x7f52cbc79060, 0x20, 0, 0x53053053) = 0

```

```

mprotect(0x7f52cbe7a000, 16384, PROT_READ) = 0
mprotect(0x55fa65398000, 4096, PROT_READ) = 0
mprotect(0x7f52cbecb000, 8192, PROT_READ) = 0
prlimit64(0,          RLIMIT_STACK,          NULL,          {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f52cbe8d000, 22943)          = 0
getrandom("\xa9\x26\x95\x73\x1f\x7c\x41\x91", 8, GRND_NONBLOCK) = 8
brk(NULL)                                = 0x55fa800fd000
brk(0x55fa8011e000)                      = 0x55fa8011e000
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x4), ...}) = 0
read(0, 1 2 3 4
"1 2 3 4\n", 1024)                      = 8
read(0, 5 6 7 8
"5 6 7 8\n", 1024)                      = 8
read(0, 9 10 11 12
"9 10 11 12\n", 1024)                  = 11
clock_gettime(CLOCK_PROCESS_CPUTIME_ID,          {tv_sec=0,
tv_nsec=1628100}) = 0
rt_sigaction(SIGRT_1,          {sa_handler=0x7f52cbd14520,          sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO,
sa_restorer=0x7f52cbcc0320}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
mmap(NULL,          8392704,          PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f52cb477000
mprotect(0x7f52cb478000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CL
ONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SET
TID|CLONE_CHILD_CLEAR_TID,          child_tid=0x7f52cbc77990,

```

```

parent_tid=0x7f52cbc77990,      exit_signal=0,      stack=0x7f52cb477000,
stack_size=0x7fff80, tls=0x7f52cbc776c0} => {parent_tid=[0]}, 88) = 9159
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=0x7f52cbc77990,      exit_signal=0,      stack=0x7f52cb477000,
stack_size=0x7fff80, tls=0x7f52cbc776c0} => {parent_tid=[0]}, 88) = 9160
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0,
tv_nsec=2471900}) = 0
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++

```

Системные вызовы:

pthread_create - инициализирует потоки и задает им функцию для выполнения

pthread_exit - завершает выполнение потока

pthread_join - ожидает завершения выполнения всех потоков

Lab3

```

rissochek@admin:/mnt/c/Users/rusla/coding/OSLabs/lab3$ strace ./parent testt.txt
execve("./parent", [".parent", "testt.txt"], 0x7ffd8fdec488 /* 35 vars */) = 0
brk(NULL) = 0x5561cd716000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8a66e37000
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=22943, ...}) = 0
mmap(NULL, 22943, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8a66e31000

```



```

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\3\0>\0\1\0\0\0\220\243\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
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7f8a66c1f000
mmap(0x7f8a66c47000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7f8a66c47000
mmap(0x7f8a66dcf000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) =
0x7f8a66dcf000
mmap(0x7f8a66e1e000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) =
0x7f8a66e1e000
mmap(0x7f8a66e24000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8a66e24000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8a66c1c000
arch_prctl(ARCH_SET_FS, 0x7f8a66c1c740) = 0
set_tid_address(0x7f8a66c1ca10) = 10902
set_robust_list(0x7f8a66c1ca20, 24) = 0
rseq(0x7f8a66c1d060, 0x20, 0, 0x53053053) = 0

```

```

mprotect(0x7f8a66e1e000, 16384, PROT_READ) = 0
mprotect(0x5561abe37000, 4096, PROT_READ) = 0
mprotect(0x7f8a66e6f000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f8a66e31000, 22943) = 0
getrandom("\x9a\x73\x4d\x80\x93\x04\x64\x25", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x5561cd716000
brk(0x5561cd737000) = 0x5561cd737000
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x4), ...}) = 0
read(0, 2 2
"2 2\n", 1024) = 4
openat(AT_FDCWD, "/dev/shm/Lab3.back",
O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0644) = 3
ftruncate(3, 50) = 0
mmap(NULL, 60, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) =
0x7f8a66e36000
clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f8a66c1ca10) = 10927
wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 10927
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=10927,
si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---
exit_group(0) = ?
+++ exited with 0 +++

```

Системные вызовы

1. **shm_open** - создаёт/открывает объекты общей памяти POSIX.
2. **ftruncate** - обрезает файл до заданного размера.
3. **mmap, munmap** - отображает файлы или устройства в памяти, или удаляет их отображение.

4. **memset** - заполнение памяти значением определённого байта.
5. **close** - закрывает файловый дескриптор.
6. **execl** - запуск файла на исполнение.
7. **perror** – вывод сообщения об ошибке.
8. **exit** - завершает выполнение программы.
9. **wait** - получает статус завершения дочернего процесса.
10. **close** - закрывает файл, а также файловые дескрипторы.

Lab4

```

rissochek@admin:/mnt/c/Users/rusla/coding/OSLabs/lab4$ strace ./first 1 2 3 2 5 6
execve("./first", ["/first", "1", "2", "3", "2", "5", "6"], 0x7fff3379f0a0 /* 35 vars */)
= 0
brk(NULL) = 0x558e00ed6000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f366a451000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/libs/First_lib.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"... , 832) =
832
fstat(3, {st_mode=S_IFREG|0777, st_size=15568, ...}) = 0
getcwd("/mnt/c/Users/rusla/coding/OSLabs/lab4", 128) = 38
mmap(NULL, 16408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)
= 0x7f366a44c000
mmap(0x7f366a44d000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7f366a44d000
mmap(0x7f366a44e000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f366a44e000

```

```

mmap(0x7f366a44f000,          8192,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    3,    0x2000)    =
0x7f366a44f000
close(3)                      = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=22943, ...}) = 0
mmap(NULL, 22943, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f366a446000
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\3\0>\0\1\0\0\0\220\243\2\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"...,
784, 64) = 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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"...,
784, 64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7f366a234000
mmap(0x7f366a25c000,          1605632,          PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    3,    0x28000)    =
0x7f366a25c000
mmap(0x7f366a3e4000,          323584,          PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    3,    0x1b0000)    =
0x7f366a3e4000
mmap(0x7f366a433000,          24576,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    3,    0x1fe000)    =
0x7f366a433000
mmap(0x7f366a439000,          52624,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f366a439000

```

```

close(3)                                = 0
mmap(NULL,                                12288,                                PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f366a231000
arch_prctl(ARCH_SET_FS, 0x7f366a231740) = 0
set_tid_address(0x7f366a231a10)        = 11648
set_robust_list(0x7f366a231a20, 24)    = 0
rseq(0x7f366a232060, 0x20, 0, 0x53053053) = 0
mprotect(0x7f366a433000, 16384, PROT_READ) = 0
mprotect(0x7f366a44f000, 4096, PROT_READ) = 0
mprotect(0x558df04c1000, 4096, PROT_READ) = 0
mprotect(0x7f366a489000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f366a446000, 22943)          = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x4), ...}) = 0
getrandom("\xdd\x27\x1b\x9d\x7b\xc1\x6c\x51", 8, GRND_NONBLOCK) = 8
brk(NULL)                              = 0x558e00ed6000
brk(0x558e00ef7000)                    = 0x558e00ef7000
write(1, "Function is evclid_gfc\n", 23Function is evclid_gfc
) = 23
write(1, "Evclid algorithm result: 1\n", 27Evclid algorithm result: 1
) = 27
write(1, "Function is square_square\n", 26Function is square_square
) = 26
write(1, "Stupid algorithm result: 30.0000"..., 35Stupid algorithm result:
30.000000
) = 35
exit_group(0)                          = ?
+++ exited with 0 +++
rissochek@admin:/mnt/c/Users/rusla/coding/OSLabs/lab4$ strace ./second

```

```

execve("./second", ["/second"], 0x7ffec27cdd00 /* 35 vars */) = 0
brk(NULL) = 0x55befddcc000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f75cef79000
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=22943, ...}) = 0
mmap(NULL, 22943, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f75cef73000
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\3\0>\0\1\0\0\0\220\243\2\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"..., 784, 64) = 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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"..., 784, 64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f75ced61000
mmap(0x7f75ced89000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f75ced89000
mmap(0x7f75cef11000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7f75cef11000
mmap(0x7f75cef60000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7f75cef60000

```

```

mmap(0x7f75cef66000,          52624,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f75cef66000
close(3)                        = 0
mmap(NULL,                    12288,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f75ced5e000
arch_prctl(ARCH_SET_FS, 0x7f75ced5e740) = 0
set_tid_address(0x7f75ced5ea10)      = 11695
set_robust_list(0x7f75ced5ea20, 24)   = 0
rseq(0x7f75ced5f060, 0x20, 0, 0x53053053) = 0
mprotect(0x7f75cef60000, 16384, PROT_READ) = 0
mprotect(0x55bee6e0e000, 4096, PROT_READ) = 0
mprotect(0x7f75cefb1000, 8192, PROT_READ) = 0
prlimit64(0,          RLIMIT_STACK,          NULL,          {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f75cef73000, 22943)          = 0
getrandom("\xab\x75\x4d\xfl\x89\xde\x54\xc7", 8, GRND_NONBLOCK) = 8
brk(NULL)                        = 0x55befddcc000
brk(0x55befddded000)             = 0x55befddded000
openat(AT_FDCWD, "./libs/First_lib.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"... , 832) =
832
fstat(3, {st_mode=S_IFREG|0777, st_size=15568, ...}) = 0
getcwd("/mnt/c/Users/rusla/coding/OSLabs/lab4", 128) = 38
mmap(NULL, 16408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)
= 0x7f75cef74000
mmap(0x7f75cef75000,          4096,          PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,      3,      0x1000)      =
0x7f75cef75000

```

```

mmap(0x7f75cef76000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f75cef76000
mmap(0x7f75cef77000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f75cef77000
close(3) = 0
mprotect(0x7f75cef77000, 4096, PROT_READ) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x4), ...}) = 0
read(0, 1 2 3
"1 2 3\n", 1024) = 6
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x4), ...}) = 0
write(1, "Function is evclid_gfc\n", 23Function is evclid_gfc
) = 23
write(1, "1\n", 21
) = 2
read(0, 2 5 6
"2 5 6\n", 1024) = 6
write(1, "Function is square_square\n", 26Function is square_square
) = 26
write(1, "30.000000\n", 1030.000000
) = 10
read(0, 0
"0\n", 1024) = 2
munmap(0x7f75cef74000, 16408) = 0
openat(AT_FDCWD, "./libs/Second_lib.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0777, st_size=15568, ...}) = 0
getcwd("/mnt/c/Users/rusla/coding/OSLabs/lab4", 128) = 38

```



```

mmap(NULL, 16408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)
= 0x7f75cef74000
mmap(0x7f75cef75000,          4096,          PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,      3,      0x1000)      =
0x7f75cef75000
mmap(0x7f75cef76000,          4096,          PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,      3,      0x2000)      =
0x7f75cef76000
mmap(0x7f75cef77000,          8192,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,      3,      0x2000)      =
0x7f75cef77000
close(3)                      = 0
mprotect(0x7f75cef77000, 4096, PROT_READ) = 0
write(1, "contract has been changed\n", 26contract has been changed
) = 26
write(1, "contract is second_realization\n", 31contract is second_realization
) = 31
read(0, 1 2 3
"1 2 3\n", 1024)              = 6
write(1, "Function is stupid_gfc\n", 23Function is stupid_gfc
) = 23
write(1, "1\n", 21
) = 2
read(0, 2 5 6
"2 5 6\n", 1024)              = 6
write(1, "Function is square_triangle\n", 28Function is square_triangle
) = 28
write(1, "15.000000\n", 1015.000000
) = 10
exit_group(0)                  = ?

```

+++ exited with 0 +++

Системные вызовы:

<code>void *dlopen(const char *filename, int flag);</code>	Загружает динамическую библиотеку, имя которой указано в строке filename и возвращает прямой указатель на начало загруженной библиотеки.
<code>const char *dlerror(void);</code>	Возвращает указатель на начало строки, описывающей ошибку, полученную на предыдущем вызове.
<code>void *dlsym(void *handle, char *symbol);</code>	Получает параметр handle, который является выходом вызова dlopen и параметр symbol, который является строкой, в которой содержится название символа, который необходимо загрузить из библиотеки. Возвращает указатель на область памяти, в которой содержится необходимый символ.
<code>int dlclose(void *handle);</code>	Уменьшает счетчик ссылок на указатель handle и если он равен нулю, то освобождает библиотеку.

Lab5-7

Системные вызовы

<code>zsock_t* zsock_new_pub(const char* endpoint)</code>	Создает новый ZMQ_PUB сокет
<code>zpoller_t* zpoller_new(void* reader)</code>	Создает новый poller, проверяющий наличие сообщений от ZMQ_PUB, на который узел подписан
<code>void* zpoller_wait(zpoller_t* self, timeout)</code>	Проверяет наличие сообщений в неблокирующем режиме
<code>char* zstr_recv (void *source);</code>	Принимает сообщение от другого сокета
<code>int zpoller_add (zpoller_t *self, void *reader);</code>	Добавляет к poller-у сокет для отслеживания
<code>int zsock_connect (zsock_t *self, const char *format, ...)</code>	Соединяет сокет с другим сокетом по адресу
<code>int zstr_send (void *dest, const char *string)</code>	Отправляет строку от нашего сокета

Выводы

Проделав работу, я приобрел практические навыки, необходимые для работы с утилитой strace, а также повторил все виды системных

вызовов, использованных в лабораторных работах, сделанных в течение курса.