

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: Шароченкова Софья Максимовна
Группа: М8О-209Б-23
Вариант: 22
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2024

Содержание

- Репозиторий
- Постановка задачи
- Демонстрация работы программы
- Выводы

Репозиторий

<https://github.com/Terentiyy/OS/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.

Демонстрация работы программы

Лабораторная работа №1

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 из файла с файловым дескриптором в область памяти
7. **write** - записывает в файл с файловым дескриптором из области памяти количество байт.

8. **pererror** – вывод сообщения об ошибке.

9. **exit** - завершает выполнение программы.

10. **wait** - получает статус завершения дочернего процесса.

```
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\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) = 0x7fbb7ae000
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(0x7fbb7ad000, 16384, PROT_READ) = 0
mprotect(0x55f974d44000, 4096, PROT_READ) = 0
mprotect(0x7fbb7afe000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7fbb7c0000, 22943) = 0
getrandom("\x8b\x96\x63\xaf\x4c\x14\xe5\xe", 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_etime=0, si_stime=0} ---
exit_group(0) = ?
+++ exited with 0 +++

```

Лабораторная работа №2

Системные вызовы:

1. **execve** – запуск исполняемого файла.

2. **mmap** – выделение памяти.

3. **openat** – открытие файлов.

4. **write** – вывод на экран.

5. **exit_group** – завершение процесса.

```
execve("./lab2", ["/lab2"], 0x7ffda9f13f60 /* 44 vars */) = 0
brk(NULL) = 0x5c0b1d19c000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fffb173de50) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x701835097000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=60907, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 60907, PROT_READ, MAP_PRIVATE, 3, 0) = 0x701835088000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\3\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\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\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0"..., 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 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"..., 784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x701834e00000
mprotect(0x701834e28000, 2023424, PROT_NONE) = 0
mmap(0x701834e28000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x701834e28000
mmap(0x701834fbd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x701834fbd000
mmap(0x701835016000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x701835016000
mmap(0x70183501c000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x70183501c000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x701835085000
arch_prctl(ARCH_SET_FS, 0x701835085740) = 0
```

```

set_tid_address(0x701835085a10)      = 5037
set_robust_list(0x701835085a20, 24)   = 0
rseq(0x7018350860e0, 0x20, 0, 0x53053053) = 0
mprotect(0x701835016000, 16384, PROT_READ) = 0
mprotect(0x5c0b1b386000, 4096, PROT_READ) = 0
mprotect(0x7018350d1000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x701835088000, 60907)         = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
getrandom("\xa5\xc5\x09\xf2\x1d\xa7\xc4\xd8", 8, GRND_NONBLOCK) = 8
brk(NULL)                             = 0x5c0b1d19c000
brk(0x5c0b1d1bd000)                   = 0x5c0b1d1bd000
write(1, "Usage: ./lab2 <array size> <numb"... , 47Usage: ./lab2 <array size> <number of threads>
) = 47
exit_group(1)                         = ?
+++ exited with 1 +++

```

Лабораторная работа №3

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.

```
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"...,

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"...,

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"...,

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(0x55561abe37000, 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) = 0x55561cd716000
brk(0x55561cd737000) = 0x55561cd737000
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_CLEAR_TID|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} ---

```

```
+++ exited with 0 +++
```

Системные вызовы:

1. **execve** – запуск исполняемого файла.
2. **mmap** – выделение памяти.
3. **write** – запись результатов.
4. **openat** – открытие библиотек.
5. **read** – чтение данных.
6. **munmap** - освобождает диапазон памяти, который был предыдущим образом отображен с помощью mmap.
7. **fstat** - получает статус файла, используя его дескриптор.
8. **access** - проверяет доступность файла по заданному пути с указанными правами.
9. **close** - закрывает указанный файловый дескриптор.

```
execve("./main", ["/.main"], 0x7ffffb25470 /* 28 vars */) = 0
brk(NULL) = 0x7ffffbe01000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
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=42278, ...}) = 0
mmap(NULL, 42278, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f0cd1c5d000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
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\0P\16\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=14560, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0cd1c50000
mmap(NULL, 2109712, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0cd17f0000
mprotect(0x7f0cd17f3000, 2093056, PROT_NONE) = 0
mmap(0x7f0cd19f2000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f0cd19f2000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libstdc++.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\220\304\10\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1594864, ...}) = 0
mmap(NULL, 3702848, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0cd1460000
mprotect(0x7f0cd15d9000, 2097152, PROT_NONE) = 0
```

```

mmap(0x7f0cd17d9000, 49152, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x179000) = 0x7f0cd17d9000
mmap(0x7f0cd17e5000, 12352, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)
= 0x7f0cd17e5000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", 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\300*\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=96616, ...}) = 0
mmap(NULL, 2192432, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0cd1240000
mprotect(0x7f0cd1257000, 2093056, PROT_NONE) = 0
mmap(0x7f0cd1456000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x16000) = 0x7f0cd1456000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
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\20\35\2\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=2030928, ...}) = 0
mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0cd0e40000
mprotect(0x7f0cd1027000, 2097152, PROT_NONE) = 0
mmap(0x7f0cd1227000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1e7000) = 0x7f0cd1227000
mmap(0x7f0cd122d000, 15072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)
= 0x7f0cd122d000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.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\200\272\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1700792, ...}) = 0
mmap(NULL, 3789144, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0cd0aa0000
mprotect(0x7f0cd0c3d000, 2093056, PROT_NONE) = 0
mmap(0x7f0cd0e3c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x19c000) = 0x7f0cd0e3c000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0cd1c40000
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0cd1c30000
arch_prctl(ARCH_SET_FS, 0x7f0cd1c30740) = 0
mprotect(0x7f0cd1227000, 16384, PROT_READ) = 0
mprotect(0x7f0cd0e3c000, 4096, PROT_READ) = 0
mprotect(0x7f0cd1456000, 4096, PROT_READ) = 0
mprotect(0x7f0cd17d9000, 40960, PROT_READ) = 0
mprotect(0x7f0cd19f2000, 4096, PROT_READ) = 0
mprotect(0x7f0cd2002000, 4096, PROT_READ) = 0
mprotect(0x7f0cd1c29000, 4096, PROT_READ) = 0
munmap(0x7f0cd1c5d000, 42278) = 0
brk(NULL) = 0x7ffffbeecd1000
brk(0x7ffffbecf2000) = 0x7ffffbecf2000
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
write(1, "0 for change libs, 1 for re1.cpp"... , 61) = 61
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
read(0, "1\n", 1024) = 2
openat(AT_FDCWD, "./libd1.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\340\4\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=7568, ...}) = 0
mmap(NULL, 2101296, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0cd0890000
mprotect(0x7f0cd0891000, 2093056, PROT_NONE) = 0
mmap(0x7f0cd0a90000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) =
0x7f0cd0a90000
close(3) = 0
mprotect(0x7f0cd0a90000, 4096, PROT_READ) = 0
write(1, "1 for GCF, 2 for Sort:\n", 23) = 23
read(0, "1 8 4\n", 1024) = 6
write(1, "4\n", 2) = 2
read(0, "2 1 4 3 2\n", 1024) = 10

```

```

write(1, "1 2 3 4 \n", 9)           = 9
read(0, "0\n", 1024)                 = 2
munmap(0x7f0cd0890000, 2101296)      = 0
openat(AT_FDCWD, "./libd2.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\5\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=7648, ...}) = 0
getcwd("/home/papey/Ubuntu/OS/os_lab5/src", 128) = 34
mmap(NULL, 2101304, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0cd0890000
mprotect(0x7f0cd0891000, 2093056, PROT_NONE) = 0
mmap(0x7f0cd0a90000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0x7f0cd0a90000
close(3)                             = 0
mprotect(0x7f0cd0a90000, 4096, PROT_READ) = 0
write(1, "re1 changed on re2\n", 19) = 19
read(0, "1 8 12\n", 1024)            = 7
write(1, "4\n", 2)                   = 2
read(0, "2 7 6 5 4 3 2 1 -1\n", 1024) = 19
write(1, "-1 1 2 3 4 5 6 7 \n", 18)  = 18
read(0, "3\n", 1024)                 = 2
munmap(0x7f0cd0890000, 2101304)      = 0
lseek(0, -1, SEEK_CUR)                = -1 EPIPE (Illegal seek)
exit_group(0)                         = ?
+++ exited with 0 +++

```

Лабораторные работы №5-7

Системные вызовы:

1. **execve** – запуск исполняемого файла.

```
execv("./client", args);
```

2. **mmap** – выделение памяти.

3. **close** – закрытие файловых дескрипторов.

4. **write** – запись результатов.

5. **openat** – открытие библиотек.

6. **read** – чтение данных.

7. **strdup** - дублирует строку и возвращает указатель на новую строку в динамической памяти.

```
char* arg0 = strdup("./client");
```

8. **bind** - привязывает сокет к указанному адресу и порту, позволяя ему слушать входящие соединения.

```
main_socket.bind(get_port_name(PORT_BASE + node_id));
```

9. **stoi** - преобразует строку в целое число, если это возможно.

```
node_id = stoi(str);
```

10. **fork** - создает новый процесс, который является копией текущего процесса.

```
child_pid = fork();
```

11. **send** - отправляет сообщение через сокет.

```
return socket.send(message);
```

12. **recv** - принимает сообщение через сокет.

```
ok = socket.recv(&message);
```

Выводы

В процессе выполнения лабораторной работы с утилитой strace я ознакомилась с ее функционалом, который позволяет отслеживать системные вызовы и исследовать поведение процессов в реальном времени. Благодаря этому опыту я смогла глубже понять, как функционируют внутренние механизмы операционной системы на низком уровне. В частности, изучение выводов strace дало мне возможность проанализировать, какие именно вызовы и операции происходят при выполнении программ, а также как они взаимодействуют с файловой системой и управляют памятью.

Кроме того, я научилась эффективно диагностировать ошибки и выявлять потенциальные проблемы в процессе исполнения приложений, что значительно ускоряет процесс отладки. Этот опыт подтвердил важность понимания системных вызовов, поскольку изучение их работы способствует более глубокому пониманию принципов и архитектуры операционной системы, что, в свою очередь, позволяет создавать более надежные и производительные программные решения. В целом, практика с утилитой strace представила собой ценное дополнение к моим знаниям и навыкам в области программирования.