

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: Чернуос Алексей Тимофеевич
Группа: М8О-209Б-23
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2024

Содержание

Репозиторий	3
Постановка задачи.....	3
Выполнение задания	3
Выводы	22

Репозиторий

https://github.com/HGRaicer/MAI_OS

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

```
gcc -o build/child src/child.c
strace ./build/parent
execve("./build/parent", ["/build/parent"], 0x7ffe68047690 /* 64 vars */) = 0
brk(NULL)                               = 0x55f590489000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff9cd9760) = -1 EINVAL
(Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7facc960c000
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=75267, ...},
AT_EMPTY_PATH) = 0
mmap(NULL, 75267, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7facc95f9000
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\0P\237\2\0\0\0\0"..., 832)
= 832
pread64(3, "\6\0\0\04\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\05\0\0\0GNU\02\0\0300\4\0\0\03\0\0\0\0\0\0"..., 48,
848) = 48
```

```

pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\302\211\332Pq\2439\235\350\223\322\257\201\3
26\243\4f"..., 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"...,
784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7facc93d0000
mprotect(0x7facc93f8000, 2023424, PROT_NONE) = 0
mmap(0x7facc93f8000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7facc93f8000
mmap(0x7facc958d000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) =
0x7facc958d000
mmap(0x7facc95e6000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7facc95e6000
mmap(0x7facc95ec000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7facc95ec000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7facc93cd000
arch_prctl(ARCH_SET_FS, 0x7facc93cd740) = 0
set_tid_address(0x7facc93cda10) = 17493
set_robust_list(0x7facc93cda20, 24) = 0
rseq(0x7facc93ce0e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7facc95e6000, 16384, PROT_READ) = 0
mprotect(0x55f590093000, 4096, PROT_READ) = 0
mprotect(0x7facc9646000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7facc95f9000, 75267) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
getrandom("\xd7\xdd\x39\xa9\x9f\xdb\x6a\x7b", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x55f590489000
brk(0x55f5904aa000) = 0x55f5904aa000
write(1, "\320\237\321\203\321\202\321\214 \320\264\320\273\321\217
\321\204\320\260\320\271\320\273\320\260 \320\262\321\213\320"..., 41Путь для
файла вывода:

```

```

) = 41
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
read(0, output
"output\n", 1024)          = 7
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\207\320\270\321\201\320\273\320\260, \321\200\320\260\320"..., 70Введите
числа, разделенные пробелами:
) = 70
read(0, 2 3 4 5 6
"2 3 4 5 6\n", 1024)      = 10
pipe2([3, 4], 0)          = 0
clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7facc93cda10) = 17578
close(3)                   = 0
write(4, "output\0\0\0\0\0\0\254\177\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\377\177\0\0"...,
512) = 512
write(4, "|", 1)           = 1
write(4, "2 3 4 5 6\n\0\0\0\0\0\0\0\0\0\0\0\0\377\0\0\0\0\0\0\0\0\0\0\0\0"..., 512) = 512
close(4)                   = 0
wait4(-1, Данные успешно записаны в файл: output
[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 17578
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=17578,
si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---
exit_group(0)              = ?
+++ exited with 0 +++

```

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

```
rai@rai-laptop:~/cods/os/lab2$ strace ./a.out 4 10 0 0 10
execve("./a.out", ["/a.out", "4", "10", "0", "0", "10"], 0x7ffd499daac8 /* 59 vars */)
= 0
brk(NULL)                               = 0x559afb531000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd67fc8a00) = -1 EINVAL
(Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdc4886e000
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=75267, ...},
AT_EMPTY_PATH) = 0
mmap(NULL, 75267, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fdc4885b000
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\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"...,
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\0\302\211\332Pq\2439\235\350\223\322\257\201\3
26\243\4f"... , 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\0"... ,
784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7fdc48632000
mprotect(0x7fdc4865a000, 2023424, PROT_NONE) = 0
mmap(0x7fdc4865a000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7fdc4865a000
mmap(0x7fdc487ef000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) =
0x7fdc487ef000
mmap(0x7fdc48848000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7fdc48848000
mmap(0x7fdc4884e000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fdc4884e000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdc4862f000
arch_prctl(ARCH_SET_FS, 0x7fdc4862f740) = 0
set_tid_address(0x7fdc4862fa10) = 17714
set_robust_list(0x7fdc4862fa20, 24) = 0
rseq(0x7fdc486300e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7fdc48848000, 16384, PROT_READ) = 0
mprotect(0x559afa93d000, 4096, PROT_READ) = 0

```

```

mprotect(0x7fdc488a8000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7fdc4885b000, 75267) = 0
getrandom("\x9d\xc9\x9b\x4d\x24\x0f\x6f\x88", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x559afb531000
brk(0x559afb552000) = 0x559afb552000
clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0,
tv_nsec=1389944}) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7fdc486c3870, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO,
sa_restorer=0x7fdc48674520}, 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) = 0x7fdc47e2e000
mprotect(0x7fdc47e2f000, 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_CLEARTID, child_tid=0x7fdc4862e910,
parent_tid=0x7fdc4862e910, exit_signal=0, stack=0x7fdc47e2e000,
stack_size=0x7fff00, tls=0x7fdc4862e640} => {parent_tid=[17715]}, 88) = 17715
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc3f7ff000
Thread ID: 140584083973696
mprotect(0x7fdc3f800000, 8388608, PROT_READ|PROT_WRITE) = 0
futex(0x7fdc488aaa98, FUTEX_WAIT_PRIVATE, 2, NULL) = 0
futex(0x7fdc488aaa98, FUTEX_WAKE_PRIVATE, 1) = 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=0x7fdc3ffff910,
child_tid=0x7fdc3ffff910,
exit_signal=0,
stack=0x7fdc3f7ff000,
stack_size=0x7fff00, tls=0x7fdc3ffff640} Thread ID: 140583943272000
```

```
=> {parent_tid=[17716]}, 88) = 17716
```

```
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
```

```
mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc4760d000
mprotect(0x7fdc4760e000, 8388608, PROT_READ|PROT_WRITE) = 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=0x7fdc47e0d910,
child_tid=0x7fdc47e0d910,
exit_signal=0,
stack=0x7fdc4760d000,
stack_size=0x7fff00, tls=0x7fdc47e0d640} => {parent_tid=[17717]}, 88) = 17717
```

```
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
```

```
mmap(NULL, 8392704, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc46e0c000
mprotect(0x7fdc46e0d000, 8388608, PROT_READ|PROT_WRITE) Thread ID:
140584075449920
```

```
) = 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=0x7fdc4760c910,
child_tid=0x7fdc4760c910,
exit_signal=0,
stack=0x7fdc46e0c000,
stack_size=0x7fff00, tls=0x7fdc4760c640} Thread ID: 140584067057216
```

```
=> {parent_tid=[17718]}, 88) = 17718
```

```

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
clock_gettime(CLOCK_PROCESS_CPUTIME_ID,          {tv_sec=0,
tv_nsec=2626420}) = 0
write(1, "Player 1 wins: 8\n", 17Player 1 wins: 8
)      = 17
write(1, "Player 2 wins: 2\n", 17Player 2 wins: 2
)      = 17
write(1, "Draw: 0\n", 8Draw: 0
)      = 8
write(1, "Del time 1237.000000\n", 21Del time 1237.000000
) = 21
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

```

rai@rai-laptop:~/cods/os/lab3$ make
mkdir -p build
gcc -Wall -g -o build/parent src/parent.c
mkdir -p build
gcc -Wall -g -o build/child src/child.c
rai@rai-laptop:~/cods/os/lab3$ strace ./parent
strace: Can't stat './parent': Нет такого файла или каталога
rai@rai-laptop:~/cods/os/lab3$ strace ./build/parent out
execve("./build/parent", ["/build/parent", "out"], 0x7ffd35877b78 /* 59 vars */) = 0
brk(NULL)          = 0x56369e865000

```

```

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd42d31880) = -1 EINVAL
(Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f73ed2c1000
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=75267, ...},
AT_EMPTY_PATH) = 0
mmap(NULL, 75267, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f73ed2ae000
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\0P\237\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"...,
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\0\302\211\332Pq\2439\235\350\223\322\257\201\3
26\243\ff"..., 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"...,
784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7f73ed085000
mprotect(0x7f73ed0ad000, 2023424, PROT_NONE) = 0

```

```

mmap(0x7f73ed0ad000,          1658880,          PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    3,    0x28000)    =
0x7f73ed0ad000
mmap(0x7f73ed242000,          360448,          PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    3,    0x1bd000)    =
0x7f73ed242000
mmap(0x7f73ed29b000,          24576,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,    3,    0x215000)    =
0x7f73ed29b000
mmap(0x7f73ed2a1000,          52816,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f73ed2a1000
close(3)                      = 0
mmap(NULL,          12288,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f73ed082000
arch_prctl(ARCH_SET_FS, 0x7f73ed082740) = 0
set_tid_address(0x7f73ed082a10)    = 17892
set_robust_list(0x7f73ed082a20, 24) = 0
rseq(0x7f73ed0830e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f73ed29b000, 16384, PROT_READ) = 0
mprotect(0x56369cb45000, 4096, PROT_READ) = 0
mprotect(0x7f73ed2fb000, 8192, PROT_READ) = 0
prlimit64(0,          RLIMIT_STACK,          NULL,          {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f73ed2ae000, 75267)      = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
getrandom("\x06\x80\x29\x7f\xf3\x89\x55\xab", 8, GRND_NONBLOCK) = 8
brk(NULL)                      = 0x56369e865000
brk(0x56369e886000)            = 0x56369e886000

```

```

newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
write(1, "Enter numbers separated by space"..., 35Enter numbers separated by
spaces: ) = 35
read(0, 2 3 4 5 6
"2 3 4 5 6\n", 1024)      = 10
openat(AT_FDCWD,                      "/dev/shm/Lab3.back",
O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0644) = 3
ftruncate(3, 14)              = 0
mmap(NULL, 14, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) =
0x7f73ed2fa000
clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f73ed082a10) = 17909
wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 17909
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=17909,
si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---
munmap(0x7f73ed2fa000, 14)      = 0
close(3)                        = 0
write(1, "Child process completed successf"..., 38Child process completed
successfully.
) = 38
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. **exec1** - 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

```
rai@rai-laptop:~/cods/os/lab4$ strace ./build/dynamic
execve("./build/dynamic", [".build/dynamic"], 0x7ffd7ff155e0 /* 59 vars */) = 0
brk(NULL)                               = 0x5651174fc000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff89427bc0) = -1 EINVAL
(Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f30e023c000
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=75267, ...},
AT_EMPTY_PATH) = 0
mmap(NULL, 75267, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f30e0229000
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\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"...,
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\0\302\211\332Pq\2439\235\350\223\322\257\201\3
26\243\4f"... , 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\0"... ,
784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7f30e0000000
mprotect(0x7f30e0028000, 2023424, PROT_NONE) = 0
mmap(0x7f30e0028000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7f30e0028000
mmap(0x7f30e01bd000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) =
0x7f30e01bd000
mmap(0x7f30e0216000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7f30e0216000
mmap(0x7f30e021c000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f30e021c000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f30dffffd000
arch_prctl(ARCH_SET_FS, 0x7f30dffffd740) = 0
set_tid_address(0x7f30dffffda10) = 18009
set_robust_list(0x7f30dffffda20, 24) = 0
rseq(0x7f30dffffe0e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f30e0216000, 16384, PROT_READ) = 0
mprotect(0x565115e74000, 4096, PROT_READ) = 0

```

```

mprotect(0x7f30e0276000, 8192, PROT_READ) = 0
prlimit64(0,          RLIMIT_STACK,          NULL,          {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f30e0229000, 75267)          = 0
getrandom("\xae\x36\x90\x09\x0b\x44\x60\x87", 8, GRND_NONBLOCK) = 8
brk(NULL)                                = 0x5651174fc000
brk(0x56511751d000)                    = 0x56511751d000
openat(AT_FDCWD, "./build/lib1.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\0\0\0\0\0\0"..., 832) =
832
newfstatat(3,      "",      {st_mode=S_IFREG|0775,      st_size=15672,      ...},
AT_EMPTY_PATH) = 0
getcwd("/home/rai/cods/os/lab4", 128) = 23
mmap(NULL, 16448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)
= 0x7f30e0237000
mmap(0x7f30e0238000,          4096,          PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,      3,      0x1000)      =
0x7f30e0238000
mmap(0x7f30e0239000,          4096,          PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,      3,      0x2000)      =
0x7f30e0239000
mmap(0x7f30e023a000,          8192,          PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,      3,      0x2000)      =
0x7f30e023a000
close(3)                                = 0
mprotect(0x7f30e023a000, 4096, PROT_READ) = 0
openat(AT_FDCWD, "./build/lib2.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\0\0\0\0\0\0"..., 832) =
832

```



```

newfstatat(3, "", {st_mode=S_IFREG|0775, st_size=15640, ...},
AT_EMPTY_PATH) = 0
getcwd("/home/rai/cods/os/lab4", 128) = 23
mmap(NULL, 16440, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)
= 0x7f30e0232000
mmap(0x7f30e0233000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) =
0x7f30e0233000
mmap(0x7f30e0234000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f30e0234000
mmap(0x7f30e0235000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7f30e0235000
close(3) = 0
mprotect(0x7f30e0235000, 4096, PROT_READ) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
write(1, "select the library you want to u"... , 36select the library you want to use
) = 36
write(1, " 1:lib1(quicksort and sieve of e"... , 46 1:lib1(quicksort and sieve of
eratosthenes)
) = 46
write(1, " 2:lib2(bubble sort and naive pr"... , 47 2:lib2(bubble sort and naive prime
counting)
) = 47
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
read(0, 1
"1\n", 1024) = 2

```

```

write(1, "1: Count primes\n2: Sort array\n", 301: Count primes
2: Sort array
) = 30
write(1, "Choose option: ", 15Choose option: ) = 15
read(0, 1
"1\n", 1024) = 2
write(1, "Enter range A and B: ", 21Enter range A and B: ) = 21
read(0, 1 10
"1 10\n", 1024) = 5
write(1, "Prime count: 4\n", 15Prime count: 4
) = 15
munmap(0x7f30e0237000, 16448) = 0
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Недопустимая операция
смещения)
exit_group(0) = ?
+++ exited with 0 +++

```

```

rai@rai-laptop:~/cods/os/lab4$ strace ./build/static
execve("./build/static", ["/build/static"], 0x7ffd6aae0e00 /* 59 vars */) = 0
brk(NULL) = 0x561d533d2000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff77ee3590) = -1 EINVAL
(Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ff65c387000
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=75267, ...},
AT_EMPTY_PATH) = 0
mmap(NULL, 75267, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ff65c374000

```

```

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\0P\237\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 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48,
848) = 48
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\302\211\332Pq\2439\235\350\223\322\257\201\3
26\243\f"..., 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\0"...,
784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7ff65c14b000
mprotect(0x7ff65c173000, 2023424, PROT_NONE) = 0
mmap(0x7ff65c173000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7ff65c173000
mmap(0x7ff65c308000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) =
0x7ff65c308000
mmap(0x7ff65c361000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7ff65c361000
mmap(0x7ff65c367000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ff65c367000

```

```

close(3)                                = 0
mmap(NULL,                               12288,                               PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ff65c148000
arch_prctl(ARCH_SET_FS, 0x7ff65c148740) = 0
set_tid_address(0x7ff65c148a10)         = 18021
set_robust_list(0x7ff65c148a20, 24)     = 0
rseq(0x7ff65c1490e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7ff65c361000, 16384, PROT_READ) = 0
mprotect(0x561d527bb000, 4096, PROT_READ) = 0
mprotect(0x7ff65c3c1000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7ff65c374000, 75267)           = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
getrandom("\x49\x98\x36\x6b\x90\x40\xc5\x6c", 8, GRND_NONBLOCK) = 8
brk(NULL)                                = 0x561d533d2000
brk(0x561d533f3000)                     = 0x561d533f3000
write(1, "1: Count primes\n", 161: Count primes
)    = 16
write(1, "2: Sort array\n", 142: Sort array
)    = 14
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...},
AT_EMPTY_PATH) = 0
write(1, "Choose option: ", 15Choose option: )    = 15
read(0, 1
"1\n", 1024)                             = 2
write(1, "Enter range A and B: ", 21Enter range A and B: ) = 21
read(0, 1 10
"1 10\n", 1024)                         = 5

```

write(1, "Prime count: 4\n", 15Prime count: 4

) = 15

lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Недопустимая операция смещения)

exit_group(0) = ?

+++ exited with 0 +++Системные вызовы:

void *dlopen(const char *filename, int flag);	Загружает динамическую библиотеку, имя которой указано в строке filename и возвращает прямой указатель на начало загруженной библиотеки.
const char *dlerror(void);	Возвращает указатель на начало строки, описывающей ошибку, полученную на предыдущем вызове.
void *dlsym(void *handle, char *symbol);	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.
int dlclose(void *handle);	Уменьшает счетчик ссылок на указатель handle и если он равен нулю, то освобождает библиотеку.

Lab5-7

Системные вызовы

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

Выводы

Проделав работу, я приобрел практические навыки, необходимые для работы с утилитой `strace`, а также пронаблюдал все виды системных вызовов, использованных в лабораторных работах, сделанных в течение курса.