

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Компьютерные науки и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №8
По курсу «Операционные системы»

Студент: Григорьев Т. А.

Группа: М8О-208Б-23

Преподаватель: Живалев Е. А.

Дата: _____

Оценка: _____

Подпись: _____

Москва, 2024


```

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) = 0x7f47ee800000

mmap(0x7f47ee828000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENY-
WRITE, 3, 0x28000) = 0x7f47ee828000

mmap(0x7f47ee9b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7f47ee9b0000

mmap(0x7f47ee9ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1fe000) = 0x7f47ee9ff000

mmap(0x7f47eea05000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f47eea05000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f47eea26000

arch_prctl(ARCH_SET_FS, 0x7f47eea26740) = 0

set_tid_address(0x7f47eea26a10) = 17757

set_robust_list(0x7f47eea26a20, 24) = 0

rseq(0x7f47eea27060, 0x20, 0, 0x53053053) = 0

mprotect(0x7f47ee9ff000, 16384, PROT_READ) = 0

mprotect(0x5ad20b442000, 4096, PROT_READ) = 0

mprotect(0x7f47eea70000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x7f47eea29000, 59307) = 0

unlink("/dev/shm/sem.sem_read1") = 0

unlink("/dev/shm/sem.sem_read2") = 0

unlink("/dev/shm/sem.sem_write") = 0

```

```

openat(AT_FDCWD, "/dev/shm/sem.sem_read1", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

getrandom("\xa3\xee\x00\x9b\x06\xaf\xac\x82", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.B0oKWc", 0x7ffd8cdcb400, AT_SYMLINK_NOFOLLOW) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "/dev/shm/sem.B0oKWc", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC,
0666) = 3

write(3, "\0\0\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) =
32

mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f47eea37000

link("/dev/shm/sem.B0oKWc", "/dev/shm/sem.sem_read1") = 0

fstat(3, {st_mode=S_IFREG|0664, st_size=32, ...}) = 0

getrandom("\xef\x70\xd9\xbd\xdl\x2c\x37\x96", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x5ad20e196000

brk(0x5ad20e1b7000) = 0x5ad20e1b7000

unlink("/dev/shm/sem.B0oKWc") = 0

close(3) = 0

openat(AT_FDCWD, "/dev/shm/sem.sem_read2", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

getrandom("\xea\x16\x5c\x6e\x1b\x51\xf1\xc3", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.4No7af", 0x7ffd8cdcb400, AT_SYMLINK_NOFOLLOW) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "/dev/shm/sem.4No7af", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC,
0666) = 3

write(3, "\0\0\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) =
32

mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f47eea36000

link("/dev/shm/sem.4No7af", "/dev/shm/sem.sem_read2") = 0

fstat(3, {st_mode=S_IFREG|0664, st_size=32, ...}) = 0

```

```

unlink("/dev/shm/sem.sem_4No7af") = 0

close(3) = 0

openat(AT_FDCWD, "/dev/shm/sem.sem_write", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

getrandom("\x5c\x3d\xf4\x49\xe1\xee\x46\xd0", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.KId4lp", 0x7ffd8cdcb400, AT_SYMLINK_NOFOLLOW) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "/dev/shm/sem.KId4lp", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC,
0666) = 3

write(3, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) =
32

mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f47eea35000

link("/dev/shm/sem.KId4lp", "/dev/shm/sem.sem_write") = 0

fstat(3, {st_mode=S_IFREG|0664, st_size=32, ...}) = 0

unlink("/dev/shm/sem.KId4lp") = 0

close(3) = 0

openat(AT_FDCWD, "/dev/shm/memory", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0666) = 3

ftruncate(3, 1024) = 0

mmap(NULL, 1024, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f47eea34000

fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x4), ...}) = 0

read(0, out1

"out1\n", 1024) = 5

read(0, out2

"out2\n", 1024) = 5

clone(child_stack=NULL, flags=CLONE_CHILD_CLEAR_TID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f47eea26a10) = 17812

clone(child_stack=NULL, flags=CLONE_CHILD_CLEAR_TID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f47eea26a10) = 17813

```

```
read(0, fsdkljfksdjfdksljldk

"fsdkljfksdjfdksljldk\n", 1024) = 21

futex(0x7f47eea36000, FUTEX_WAKE, 1)      = 1

futex(0x7f47eea35000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY) = -1 EAGAIN (Resource temporarily unavailable)

read(0, sdlkfjlkdsjfkldsjlkfkd

"sdlkfjlkdsjfkldsjlkfkd\n", 1024) = 23

futex(0x7f47eea36000, FUTEX_WAKE, 1)      = 1

futex(0x7f47eea35000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY) = -1 EAGAIN (Resource temporarily unavailable)

read(0, sdjlfdsjlkfjsdlkfsjdklsjkd

"sdjlfdsjlkfjsdlkfsjdklsjkd\n", 1024) = 27

futex(0x7f47eea36000, FUTEX_WAKE, 1)      = 1

futex(0x7f47eea35000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY) = 0

read(0, sdfjsdnfncxnmsfd

"sdfjsdnfncxnmsfd\n", 1024)      = 17

futex(0x7f47eea36000, FUTEX_WAKE, 1)      = 1

read(0, q

"q\n", 1024)                        = 2

futex(0x7f47eea37000, FUTEX_WAKE, 1)      = 1

futex(0x7f47eea36000, FUTEX_WAKE, 1)      = 1

wait4(-1, NULL, 0, NULL)              = 17812

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=17812, si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---

wait4(-1, NULL, 0, NULL)              = 17813

munmap(0x7f47eea37000, 32)            = 0
```

```

unlink("/dev/shm/sem.sem_read1")      = 0

munmap(0x7f47eea36000, 32)             = 0

unlink("/dev/shm/sem.sem_read2")      = 0

munmap(0x7f47eea35000, 32)             = 0

unlink("/dev/shm/sem.sem_write")      = 0

munmap(0x7f47eea34000, 1024)           = 0

close(3)                               = 0

unlink("/dev/shm/memory")              = 0

exit_group(0)                          = ?

+++ exited with 0 +++

```

Использовавшиеся системные вызовы:

1. **execve** - Запускает программу, заменяя текущий процесс новым.
2. **brk** - Управляет размером кучи процесса, выделяя или освобождая память.
3. **mmap** - Отображает файл или область памяти в адресное пространство процесса, используется для выделения памяти.
4. **access** - Проверяет доступность файла для чтения, записи или выполнения.
5. **openat** - Открывает файл относительно указанной директории.
6. **fstat** - Получает информацию о файле, такую как размер и права доступа.
7. **pread64** - Читает данные из файла с указанным смещением.
8. **close** - Закрывает файловый дескриптор, освобождая ресурсы.
9. **arch_prctl** - Устанавливает значение регистра FS, используется для управления потоками.
10. **set_tid_address** - Устанавливает адрес для хранения идентификатора потока.

11. **mprotect** - Изменяет права доступа к области памяти, например, защищает её от записи.
12. **prlimit64** - Получает или устанавливает ограничения на ресурсы процесса, такие как размер стека.
13. **unlink** - Удаляет файл, используется для очистки временных файлов или семафоров.
14. **getrandom** - Генерирует случайные данные из системного генератора случайных чисел.
15. **clone** - Создает новый поток или процесс для параллельной работы.
16. **futex** - Управляет синхронизацией между потоками, например, ожиданием или пробуждением.
17. **wait4** - Ожидает завершения дочернего процесса или потока.
18. **exit_group** - Завершает выполнение всех потоков процесса, завершая программу.

Эти системные вызовы показывают, как программа взаимодействует с ядром Linux для выполнения задач, таких как управление памятью, работа с файлами, синхронизация потоков и завершение работы.

Вывод: В ходе лабораторной работы была проведена диагностика программы с использованием инструмента strace, что позволило проанализировать системные вызовы, обеспечивающие взаимодействие программы с ресурсами операционной системы. В процессе исследования были изучены механизмы работы с памятью, синхронизацией процессов, вводом-выводом и обменом сообщениями. Полученные результаты подтвердили корректность работы программы и её соответствие теоретическим ожиданиям. Анализ системных вызовов продемонстрировал их ключевую роль в обеспечении стабильного функционирования программы и её взаимодействия с операционной системой..