

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

**Лабораторная работа №1 по курсу
«Операционные системы»**

Студент: Путилин Д.Н.
Группа: М8О-207Б-21
Вариант: -
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

Содержание

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

Репозиторий

<https://github.com/putilin21dn/OC>

Постановка задачи

Цель работы

Приобретение практических навыков диагностики работы программного обеспечения.

Задание

При выполнении последующих лабораторных работ необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР 3.

По итогам выполнения всех лабораторных работ отчет по данной должен содержать краткую сводку по исследованию последующих ЛР 3.

Общие сведения о программе

`ssize_t read(int fd, void *buf, size_t count);`

`read` - чтение из файлового дескриптора

`read()` пытается считывать до подсчета байтов из файлового дескриптора `fd` в

буфер, начинающийся с `buf`.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

При успешном выполнении возвращается количество прочитанных байт (ноль указывает на конец файла), и позиция файла увеличивается на это число. Это не ошибка, если это число меньше запрошенного количества байтов;

это может произойти, например, из-за того, что на самом деле доступно меньше байтов прямо сейчас (возможно, потому, что мы были близки к завершению файла, или потому, что мы считываемся из канала или с терминала), или потому, что `read()` был прерванный сигналом.

При ошибке возвращается значение `-1`, а значение `errno` устанавливается для указания ошибки. В этом случае остается неуказанным, является ли позиция файла (если таковая имеется) изменения.

`int openat(int dirfd, const char *pathname, int flags, mode_t mode);`

`openat` - откройте и, возможно, создайте файл

Системный вызов `openat()` работает точно так же, как `open()`, за исключением описанных здесь различий.

Аргумент `dirfd` используется в сочетании с аргументом `pathname` следующим образом:

- * Если путь, указанный в `pathname`, является абсолютным, то `dirfd` - это `ignored`.
- * Если путь, указанный в `pathname`, является относительным, а `dirfd` - это `special` значение `AT_FDCWD`, тогда `pathname` интерпретируется относительно текущего рабочего каталога вызывающего процесса (например, `open()`).
- * Если путь, указанный в `pathname`, является относительным, то это `inter-` устанавливается относительно каталога, на который ссылается дескриптор файла `dirfd` (а не относительно текущего рабочего каталога вызывающего процесса, как это делается с помощью `open()` для относительного пути). В

этом случае `dirfd` должен быть каталогом, который был открыт для чтения (`O_RDONLY`) или с использованием флага `O_PATH`.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

При успешном выполнении `openat()` возвращают описание нового файла `tor` (неотрицательное целое число). При ошибке возвращается значение `-1` и устанавливается значение `errno` чтобы указать на ошибку.

int **clone3**(int (*fn)(void *), void *stack, int flags, void *arg, pid_t *parent_tid, void *tls, pid_t *child_tid);

`clone3` - создайте дочерний процесс

Системный вызов `clone3()` предоставляет дополнительный набор функциональных возможностей более старого интерфейса `clone()`. Он также предоставляет ряд улучшений API-улучшения, в том числе: пространство для дополнительных битов флагов; более чистое разделение в использовании различных аргументов; и возможность указать размер дочерней области стека.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

При успешном выполнении идентификатор потока дочернего процесса возвращается в потоке выполнения вызывающей стороны. При сбое возвращается значение `-1` в контекст вызывающего абонента, дочерний процесс не создается, и значение `errno` устанавливается в `error`.

Исходный код

report

```
dmitry@dmitry-VirtualBox:~/Рабочий стол/OC/lab3/build$ strace ./main.out
execve("./main.out", ["/main.out"], 0x7fff5127a510 /* 63 vars */) = 0
brk(NULL)                                = 0x559428492000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff7a988db0) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa02e235000
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=70013, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 70013, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fa02e223000
close(3)                                  = 0
openat(AT_FDCWD, "/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\0\0\0\0\0\0"... , 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2252064, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 2267264, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fa02de00000
mmap(0x7fa02de9a000, 1114112, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x9a000) = 0x7fa02de9a000
mmap(0x7fa02dfaa000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1aa000) = 0x7-
fa02dfaa000
mmap(0x7fa02e019000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x218000) = 0x7fa02e019000
mmap(0x7fa02e027000, 10368, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7fa02e027000
close(3) = 0
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\3\0>\0\1\0\0\0\0\0\0\0\0\0"... , 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125464, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 127688, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fa02e203000
mmap(0x7fa02e206000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) =
0x7fa02e206000
mmap(0x7fa02e21d000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7-
fa02e21d000
mmap(0x7fa02e221000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000)
= 0x7fa02e221000
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\0\0\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\0\0\0"... , 784, 64) = 784
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2072888, ...}, 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\0\0\0"... , 784, 64) = 784
mmap(NULL, 2117488, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fa02da00000
mmap(0x7fa02da22000, 1544192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x22000) = 0x7fa02da22000
mmap(0x7fa02db9b000, 356352, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19b000) = 0x7-
fa02db9b000
mmap(0x7fa02dbf2000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1f1000) = 0x7fa02dbf2000
mmap(0x7fa02dbf8000, 53104, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7fa02dbf8000
close(3) = 0
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\0\0\0\0\0\0"... , 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=948816, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 950520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fa02e11a000
mmap(0x7fa02e128000, 516096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000)
= 0x7fa02e128000
mmap(0x7fa02e1a6000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8c000) = 0x7-
fa02e1a6000
mmap(0x7fa02e201000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe6000)
= 0x7fa02e201000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa02e118000
arch_prctl(ARCH_SET_FS, 0x7fa02e119400) = 0
set_tid_address(0x7fa02e1196d0) = 9579
set_robust_list(0x7fa02e1196e0, 24) = 0
rseq(0x7fa02e119d20, 0x20, 0, 0x53053053) = 0
mprotect(0x7fa02dbf2000, 16384, PROT_READ) = 0
mprotect(0x7fa02e201000, 4096, PROT_READ) = 0
mprotect(0x7fa02e221000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa02e116000
mprotect(0x7fa02e019000, 45056, PROT_READ) = 0
mprotect(0x559428175000, 4096, PROT_READ) = 0
mprotect(0x7fa02e26b000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7fa02e223000, 70013) = 0
getrandom("\xc3\xa5\x6f\x92\x79\xc0\x25\x0c", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x559428492000
brk(0x5594284b3000) = 0x5594284b3000
futexp(0x7fa02e02773c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
```

```
read(0, ^Cstrace: Process 9579 detached
<detached ...>
```

```
dmitry@dmitry-VirtualBox:~/Рабочий стол/OC/lab3/build$ strace ./main.out 4 10
execve("./main.out", ["./main.out", "4", "10"], 0x7ffe1638c310 /* 63 vars */) = 0
brk(NULL) = 0x561c8492b000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc8fe8d4c0) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe5f701a000
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=70013, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 70013, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe5f7008000
close(3) = 0
openat(AT_FDCWD, "/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\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2252064, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 2267264, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe5f6c00000
mmap(0x7fe5f6c9a000, 1114112, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7fe5f6c9a000
mmap(0x7fe5f6daa000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1aa000) = 0x7fe5f6daa000
mmap(0x7fe5f6e19000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x218000) = 0x7fe5f6e19000
mmap(0x7fe5f6e27000, 10368, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fe5f6e27000
close(3) = 0
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\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125464, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 127688, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe5f6fe8000
mmap(0x7fe5f6feb000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fe5f6feb000
mmap(0x7fe5f7002000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7fe5f7002000
mmap(0x7fe5f7006000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7fe5f7006000
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\0\3\0>\0\1\0\0\0\0\0\0\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"..., 784, 64) = 784
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2072888, ...}, 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, 2117488, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe5f6800000
mmap(0x7fe5f6822000, 1544192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7fe5f6822000
mmap(0x7fe5f699b000, 356352, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19b000) = 0x7fe5f699b000
mmap(0x7fe5f69f2000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1f1000) = 0x7fe5f69f2000
mmap(0x7fe5f69f8000, 53104, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fe5f69f8000
close(3) = 0
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\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=948816, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 950520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe5f6eff000
mmap(0x7fe5f6f0d000, 516096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7fe5f6f0d000
mmap(0x7fe5f6f8b000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8c000) = 0x7fe5f6f8b000
mmap(0x7fe5f6fe6000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe6000) = 0x7fe5f6fe6000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe5f6efd000
arch_prctl(ARCH_SET_FS, 0x7fe5f6efe400) = 0
set_tid_address(0x7fe5f6efe6d0) = 9721
set_robust_list(0x7fe5f6efe6e0, 24) = 0
rseq(0x7fe5f6efed20, 0x20, 0, 0x53053053) = 0
```

```

mprotect(0x7fe5f69f2000, 16384, PROT_READ) = 0
mprotect(0x7fe5f6fe6000, 4096, PROT_READ) = 0
mprotect(0x7fe5f7006000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe5f6efb000
mprotect(0x7fe5f6e19000, 45056, PROT_READ) = 0
mprotect(0x561c8361e000, 4096, PROT_READ) = 0
mprotect(0x7fe5f7050000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7fe5f7008000, 70013) = 0
getrandom("\x0c\xb2\xb0\x39\x7c\xa5\x6f", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x561c8492b000
brk(0x561c8494c000) = 0x561c8494c000
futex(0x7fe5f6e2773c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
read(0, 5
"5\n", 1024) = 2
read(0, 3
"3\n", 1024) = 2
read(0, 10
"10\n", 1024) = 3
read(0, 5
"5\n", 1024) = 2
read(0, 5
"5\n", 1024) = 2
read(0, 2
"2\n", 1024) = 2
read(0, 3
"3\n", 1024) = 2
read(0, 7
"7\n", 1024) = 2
read(0, 9
"9\n", 1024) = 2
read(0, 2
"2\n", 1024) = 2
read(0, 4
"4\n", 1024) = 2
read(0, 9
"9\n", 1024) = 2
read(0, 6
"6\n", 1024) = 2
read(0, 10
"10\n", 1024) = 3
read(0, 7
"7\n", 1024) = 2
read(0, 3
"3\n", 1024) = 2
read(0, 4
"4\n", 1024) = 2
read(0, 7
"7\n", 1024) = 2
read(0, 6
"6\n", 1024) = 2
read(0, 9
"9\n", 1024) = 2
read(0, 6
"6\n", 1024) = 2
read(0, 5
"5\n", 1024) = 2
read(0, 5
"5\n", 1024) = 2
read(0, 2
"2\n", 1024) = 2
read(0, 1
"1\n", 1024) = 2
read(0, 3
"3\n", 1024) = 2
read(0, 0
"0\n", 1024) = 2

```

```

rt_sigaction(SIGRT_1, {sa_handler=0x7fe5f688d1a0, sa_mask=[], sa_flags=SA_RESTORER|SA_ONSTACK|
SA_RESTART|SA_SIGINFO, sa_restorer=0x7fe5f683bcf0}, 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) = 0x7fe5f5fff000
mprotect(0x7fe5f6000000, 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_CLEAR_TID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9850]}, 88) = 9850
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe5f57fe000
mprotect(0x7fe5f57ff000, 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_CLEAR_TID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9851]}, 88) = 9851
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe5f4ffd000
mprotect(0x7fe5f4ffe000, 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_CLEAR_TID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9852]}, 88) = 9852
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe5f47fc000
mprotect(0x7fe5f47fd000, 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_CLEAR_TID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9853]}, 88) = 9853
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f67ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9850, NULL, FUTEX_BITSET_MATCH_ANY)
= 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_CLEAR_TID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9854]}, 88) = 9854
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_CLEAR_TID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9855]}, 88) = 9855
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_CLEAR_TID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9856]}, 88) = 9856
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_CLEAR_TID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9857]}, 88) = 9857
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f67ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9857, NULL, FUTEX_BITSET_MATCH_ANY)
= 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_CLEAR_TID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9858]}, 88) = 9858

```



```

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_CLEARID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9859]}, 8) = 9859
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_CLEARID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9860]}, 8) = 9860
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_CLEARID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9861]}, 8) = 9861
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_CLEARID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9862]}, 8) = 9862
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_CLEARID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9863]}, 8) = 9863
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_CLEARID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9864]}, 8) = 9864
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_CLEARID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9865]}, 8) = 9865
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f4ffc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9862, NULL, FUTEX_BITSET_MATCH_ANY)
= 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_CLEARID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9866]}, 8) = 9866
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_CLEARID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9867]}, 8) = 9867
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_CLEARID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9868]}, 8) = 9868
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_CLEARID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9869]}, 8) = 9869

```

```

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f57fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9868, NULL, FUTEX_BITSET_MATCH_ANY)
= 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_CLEARPID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9870]}, 88) = 9870
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_CLEARPID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9871]}, 88) = 9871
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_CLEARPID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9872]}, 88) = 9872
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_CLEARPID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9873]}, 88) = 9873
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f4ffc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9870, NULL, FUTEX_BITSET_MATCH_ANY)
= 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_CLEARPID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9874]}, 88) = 9874
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_CLEARPID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9875]}, 88) = 9875
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_CLEARPID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9876]}, 88) = 9876
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_CLEARPID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9877]}, 88) = 9877
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_CLEARPID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9878]}, 88) = 9878
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_CLEARPID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9879]}, 88) = 9879
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_CLEARPID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,

```

```

exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9880]}, 88) = 9880
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_CLEARPID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9881]}, 88) = 9881
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f57fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9879, NULL, FUTEX_BITSET_MATCH_ANY)
= 0
futex(0x7fe5f67ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9881, NULL, FUTEX_BITSET_MATCH_ANY)
= 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_CLEARPID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9882]}, 88) = 9882
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_CLEARPID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9883]}, 88) = 9883
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_CLEARPID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9884]}, 88) = 9884
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_CLEARPID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9885]}, 88) = 9885
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f67ff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9882, NULL, FUTEX_BITSET_MATCH_ANY)
= 0
futex(0x7fe5f57fd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9884, NULL, FUTEX_BITSET_MATCH_ANY)
= 0
futex(0x7fe5f4ffc990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9885, NULL, FUTEX_BITSET_MATCH_ANY)
= 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_CLEARPID, child_tid=0x7fe5f4ffc990, parent_tid=0x7fe5f4ffc990,
exit_signal=0, stack=0x7fe5f47fc000, stack_size=0x7fff80, tls=0x7fe5f4ffc6c0} =>
{parent_tid=[9886]}, 88) = 9886
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_CLEARPID, child_tid=0x7fe5f57fd990, parent_tid=0x7fe5f57fd990,
exit_signal=0, stack=0x7fe5f4ffd000, stack_size=0x7fff80, tls=0x7fe5f57fd6c0} =>
{parent_tid=[9887]}, 88) = 9887
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_CLEARPID, child_tid=0x7fe5f5ffe990, parent_tid=0x7fe5f5ffe990,
exit_signal=0, stack=0x7fe5f57fe000, stack_size=0x7fff80, tls=0x7fe5f5ffe6c0} =>
{parent_tid=[9888]}, 88) = 9888
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_CLEARPID, child_tid=0x7fe5f67ff990, parent_tid=0x7fe5f67ff990,
exit_signal=0, stack=0x7fe5f5fff000, stack_size=0x7fff80, tls=0x7fe5f67ff6c0} =>
{parent_tid=[9889]}, 88) = 9889
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
futex(0x7fe5f5ffe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 9888, NULL, FUTEX_BITSET_MATCH_ANY)

```

```

= 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
write(1, "Filtr narashivania\n", 20Filtr narashivania
)      = 20
write(1, "3 10 5 5 2\n", 123 10 5 5 2
)      = 12
write(1, "3 7 9 2 4\n", 113 7 9 2 4
)      = 11
write(1, "9 6 10 7 3\n", 129 6 10 7 3
)      = 12
write(1, "4 7 6 9 6\n", 114 7 6 9 6
)      = 11
write(1, "5 5 2 1 3\n", 115 5 2 1 3
)      = 11
write(1, "\nFiltr erosii\n", 16
Filtr erosii
)      = 16
write(1, "3 10 5 5 2\n", 123 10 5 5 2
)      = 12
write(1, "3 7 9 2 4\n", 113 7 9 2 4
)      = 11
write(1, "9 6 10 7 3\n", 129 6 10 7 3
)      = 12
write(1, "4 7 6 9 6\n", 114 7 6 9 6
)      = 11
write(1, "5 5 2 1 3\n", 115 5 2 1 3
)      = 11
lseek(0, -1, SEEK_CUR)          = -1 ESPIPE (Недопустимая операция смещения)
exit_group(0)                  = ?
+++ exited with 0 +++

```

Выводы

Проделав данную лабораторную работу, я приобрел навыки, необходимые для работы с strace, а также изучил системные вызовы.