Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №1 по курсу**

**«Операционные системы»**

**Тема работы “Использование утилиты strace”**

Студент: Борисов Ян Артурович

Группа: М8О-208Б-20

Вариант: -

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Демонстрация работы с утилитой strace и подробное объяснение каждого системного вызова
4. Выводы

**Репозиторий**

<https://github.com/Yannikupy/OS>

**Постановка задачи**

Задача: подробно рассказать о каждом системном вызове, отображенном утилитой strace. Разбор системных вызовов осуществляется на примере третьей лабораторной работы курса “Операционные системы”.

**Демонстрация работы с утилитой strace и подробное объяснение каждого системного вызова**

Исходный код strace:  
   
execve("./a.out", ["./a.out", "5", "2", "3", "5", "6", "4"], 0x7fffd59a4870 /\* 17 vars \*/) = 0

brk(NULL) = 0x7fffd7946000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fffdf2354b0) = -1 EINVAL (Invalid argument)

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=36464, ...}) = 0

mmap(NULL, 36464, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fe241f67000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libpthread.so.0", 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\220\201\0\0\0\0\0\0"..., 832) = 832

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=157224, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe241fa0000

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

mmap(NULL, 140408, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe241f44000

mmap(0x7fe241f4b000, 69632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7000) = 0x7fe241f4b000

mmap(0x7fe241f5c000, 20480, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7fe241f5c000

mmap(0x7fe241f61000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c000) = 0x7fe241f61000

mmap(0x7fe241f63000, 13432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe241f63000

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\360q\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\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029224, ...}) = 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

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

mmap(NULL, 2036952, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe241d50000

mprotect(0x7fe241d75000, 1847296, PROT\_NONE) = 0

mmap(0x7fe241d75000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7fe241d75000

mmap(0x7fe241eed000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7fe241eed000

mmap(0x7fe241f38000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fe241f38000

mmap(0x7fe241f3e000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe241f3e000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe241d40000

arch\_prctl(ARCH\_SET\_FS, 0x7fe241d40740) = 0

mprotect(0x7fe241f38000, 12288, PROT\_READ) = 0

mprotect(0x7fe241f61000, 4096, PROT\_READ) = 0

mprotect(0x7fe241fa9000, 4096, PROT\_READ) = 0

mprotect(0x7fe241f9d000, 4096, PROT\_READ) = 0

munmap(0x7fe241f67000, 36464) = 0

set\_tid\_address(0x7fe241d40a10) = 75

set\_robust\_list(0x7fe241d40a20, 24) = 0

rt\_sigaction(SIGRTMIN, {sa\_handler=0x7fe241f4bbf0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7fe241f593c0}, NULL, 8) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7fe241f4bc90, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7fe241f593c0}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=8192\*1024}) = 0

time(NULL) = 1639996759 (2021-12-20T13:39:19+0300)

brk(NULL) = 0x7fffd7946000

brk(0x7fffd7967000) = 0x7fffd7967000

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fe241530000

mprotect(0x7fe241531000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7fe241d2ffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[76], tls=0x7fe241d30700, child\_tidptr=0x7fe241d309d0) = 76

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fe240d20000

mprotect(0x7fe240d21000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7fe24151ffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[77], tls=0x7fe241520700, child\_tidptr=0x7fe2415209d0) = 77

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fe240510000

mprotect(0x7fe240511000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7fe240d0ffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[78], tls=0x7fe240d10700, child\_tidptr=0x7fe240d109d0) = 78

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fe23b7f0000

mprotect(0x7fe23b7f1000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7fe23bfeffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[79], tls=0x7fe23bff0700, child\_tidptr=0x7fe23bff09d0) = 79

futex(0x7fe241d309d0, FUTEX\_WAIT, 76, NULL) = 0

futex(0x7fe23bff09d0, FUTEX\_WAIT, 79, NULL) = 0

clone(child\_stack=0x7fe23bfeffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[80], tls=0x7fe23bff0700, child\_tidptr=0x7fe23bff09d0) = 80

clone(child\_stack=0x7fe240d0ffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[81], tls=0x7fe240d10700, child\_tidptr=0x7fe240d109d0) = 81

futex(0x7fe23bff09d0, FUTEX\_WAIT, 80, NULL) = 0

futex(0x7fe240d109d0, FUTEX\_WAIT, 81, NULL) = 0

write(1, "Chances of the 1st player: 0.17\n", 32) = 32

write(1, "Chances of the 2nd player: 0.83\n", 32) = 32

exit\_group(0) = ?

+++ exited with 0 +++

Разберем подробнее:  
**execve(...)** - запускает программу  
**brk(...)** - устанавливает конец сегмента данных в значение NULL  
**access(...)** - проверяет права доступа к файлу, возвращает -1 как код ошибки  
**openat(...)** - открывает файл, имеет в качестве возвращаемого значения файловый дескриптор  
**fstat(...)** - собирает информацию из файла  
**mmap(...)** - отображает файл на память  
**mprotect(...)** - контролирует доступ к памяти  
**close(...)** - закрывает файловый дескриптор  
**read(...)** - считывает из файлового дескриптора  
**arch\_prctl(...)** - устанавливает специфичное для архитектуры значение ядра  
**munmap(...)** - освобождает память, отведенную для отображения файла  
**write(...)** - пишет в консоль  
**ftruncate(...)** - увеличивает размер файла до n байтов  
**clone(...)** - создает новый процесс  
**futex(...)** - предоставляет метод ожидания, пока какое-то условие не становится успешным

**set\_tid\_address(…)** – устанавливает указатель на поток  
  
**get\_robust\_list, set\_robust\_list** – получает/устанавливает список futex.

**Выводы**В данной лабораторной работе на примере конкретного задания мы разобрали вывод утилиты strace, разобрали все команды. Пользоваться strace полезно, так как ты напрямую видишь то, как работает твоя программа, как она реагирует на команды из консоли, это может помочь в дебаггинге программы.