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: Шандрюк Пётр Николаевич

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/Peter1811/OS/tree/main/lab1

**Постановка задачи**

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

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

Исходный код strace:  
   
execve("./main", ["./main"], 0x7fffd63d0ea0 /\* 19 vars \*/) = 0

brk(NULL) = 0x7fffec5e3000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffff4547640) = -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=35808, ...}) = 0

mmap(NULL, 35808, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fdae3527000

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`\341\t\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1956992, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdae3560000

mmap(NULL, 1972224, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdae3340000

mprotect(0x7fdae33d6000, 1290240, PROT\_NONE) = 0

mmap(0x7fdae33d6000, 987136, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x96000) = 0x7fdae33d6000

mmap(0x7fdae34c7000, 299008, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x187000) = 0x7fdae34c7000

mmap(0x7fdae3511000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d0000) = 0x7fdae3511000

mmap(0x7fdae351f000, 10240, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fdae351f000

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\3\0>\0\1\0\0\0\3405\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=104984, ...}) = 0

mmap(NULL, 107592, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdae3320000

mmap(0x7fdae3323000, 73728, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fdae3323000

mmap(0x7fdae3335000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x15000) = 0x7fdae3335000

mmap(0x7fdae3339000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7fdae3339000

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) = 0x7fdae3120000

mprotect(0x7fdae3145000, 1847296, PROT\_NONE) = 0

mmap(0x7fdae3145000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7fdae3145000

mmap(0x7fdae32bd000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7fdae32bd000

mmap(0x7fdae3308000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fdae3308000

mmap(0x7fdae330e000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fdae330e000

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\300\363\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1369352, ...}) = 0

mmap(NULL, 1368336, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdae2fd1000

mmap(0x7fdae2fe0000, 684032, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xf000) = 0x7fdae2fe0000

mmap(0x7fdae3087000, 618496, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb6000) = 0x7fdae3087000

mmap(0x7fdae311e000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14c000) = 0x7fdae311e000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdae2fc0000

arch\_prctl(ARCH\_SET\_FS, 0x7fdae2fc1100) = 0

mprotect(0x7fdae3308000, 12288, PROT\_READ) = 0

mprotect(0x7fdae311e000, 4096, PROT\_READ) = 0

mprotect(0x7fdae3339000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdae2fb0000

mprotect(0x7fdae3511000, 45056, PROT\_READ) = 0

mprotect(0x7fdae356a000, 4096, PROT\_READ) = 0

mprotect(0x7fdae355d000, 4096, PROT\_READ) = 0

munmap(0x7fdae3527000, 35808) = 0

brk(NULL) = 0x7fffec5e3000

brk(0x7fffec604000) = 0x7fffec604000

pipe([3, 4]) = 0

pipe([5, 6]) = 0

pipe([7, 8]) = 0

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fdae2fc13d0) = 177

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fdae2fc13d0) = 178

close(3) = 0

close(6) = 0

close(5) = 0

close(8) = 0

fstat(0, {st\_mode=S\_IFCHR|0660, st\_rdev=makedev(0x4, 0x1), ...}) = 0

ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0

read(0, "3\n", 512) = 2

write(4, "\3\0\0\0", 4) = 4

read(0, "fgfdsg sfgsfgf\n", 512) = 15

read(0, "dfgfd \n", 512) = 10

read(0, "dfgsdfg dfsgdfsg \n", 512) = 19

read(0, "", 512) = 0

write(4, "-\0\0\0", 4) = 4

write(4, "\nfgfdsg sfgsfgf\ndfgfd \ndfgsdf"..., 45) = 45

write(4, "\0\0\0\0", 4) = 4

write(4, "", 0) = 0

write(4, "\0\0\0\0", 4) = 4

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=177, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

write(4, "", 0) = 0

read(7, "-\0\0\0", 4) = 4

read(7, "\nFGFDSG\_SFGSFGF\nDFGFD\_\_\_\_\nDFGSDF"..., 45) = 45

fstat(1, {st\_mode=S\_IFCHR|0660, st\_rdev=makedev(0x4, 0x1), ...}) = 0

ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0

write(1, "\n", 1) = 1

write(1, "FGFDSG\_SFGSFGF\n", 15) = 15

write(1, "DFGFD\_\_\_\_\n", 10) = 10

write(1, "DFGSDFG\_DFSGDFSG\_\_\n", 19) = 19

read(7, "\0\0\0\0", 4) = 4

read(7, "", 0) = 0

read(7, "\0\0\0\0", 4) = 4

read(7, "", 0) = 0

close(4) = 0

close(7) = 0

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  
unlink(...) - удаляет файл  
lseek(...) - устанавливает смещение для позиционирования операций чтения/записи

**Выводы**В данной лабораторной работе на примере конкретного задания мы разобрали вывод утилиты strace, прошлись по всем командам. Пользоваться strace очень и очень полезно, так как ты напрямую видишь то, как работает твоя программа, как она реагирует на команды из консоли.