

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: Москвин Артём
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/Pert002>

Постановка задачи

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

Демонстрация работы с утилитой strace и подробное объяснение каждого системного вызова

Исходный код strace:

```
execve("./a.out", ["/a.out"], 0x7fffd9b4d260 /* 16 vars */) = 0
brk(NULL)                               = 0x7ffff5acf000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
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=30312, ...}) = 0
mmap(NULL, 30312, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8990f6c000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/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\3\0>\0\1\0\0\0\220\304\10\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1594864, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8990f60000
mmap(NULL, 3702848, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f8990870000
mprotect(0x7f89909e9000, 2097152, PROT_NONE) = 0
mmap(0x7f8990be9000, 49152, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x179000) = 0x7f8990be9000
```

```

mmap(0x7f8990bf5000, 12352, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8990bf5000

close(3) = 0

access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)

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

fstat(3, {st_mode=S_IFREG|0644, st_size=96616, ...}) = 0

mmap(NULL, 2192432, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f8990650000

mprotect(0x7f8990667000, 2093056, PROT_NONE) = 0

mmap(0x7f8990866000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x16000) = 0x7f8990866000

close(3) = 0

access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)

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\0\20\35\2\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0755, st_size=2030928, ...}) = 0

mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f8990250000

mprotect(0x7f8990437000, 2097152, PROT_NONE) = 0

mmap(0x7f8990637000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f8990637000

mmap(0x7f899063d000, 15072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f899063d000

close(3) = 0

access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)

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\3\0>\0\1\0\0\0\200\272\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=1700792, ...}) = 0

mmap(NULL, 3789144, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f898feb0000

mprotect(0x7f899004d000, 2093056, PROT_NONE) = 0

mmap(0x7f899024c000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19c000) = 0x7f899024c000

close(3) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f8990f50000

```

```

arch_prctl(ARCH_SET_FS, 0x7f8990f50d80) = 0
mprotect(0x7f8990637000, 16384, PROT_READ) = 0
mprotect(0x7f899024c000, 4096, PROT_READ) = 0
mprotect(0x7f8990866000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f8990f40000
mprotect(0x7f8990be9000, 40960, PROT_READ) = 0
mprotect(0x7f8991203000, 4096, PROT_READ) = 0
mprotect(0x7f8990e29000, 4096, PROT_READ) = 0
munmap(0x7f8990f6c000, 30312)      = 0
brk(NULL)                        = 0x7fff5acf000
brk(0x7fff5af0000)               = 0x7fff5af0000
pipe([3, 4])                     = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f8990f51050) = 174
fstat(1, {st_mode=S_IFCHR|0660, st_rdev=makedev(4, 1), ...}) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
getpid()                         = 173
write(1, "Parent's PID: 173\n", 18)  = 18
write(1, "Child's PID: 174\n", 17)   = 17
write(1, "Enter the file name:\n", 21) = 21
fstat(0, {st_mode=S_IFCHR|0660, st_rdev=makedev(4, 1), ...}) = 0
ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0
read(0, "file\n", 512)             = 5
write(4, "\4\0\0\0", 4)             = 4
write(4, "f", 1)                   = 1
write(4, "i", 1)                   = 1
write(4, "l", 1)                   = 1
write(4, "e", 1)                   = 1
write(1, "Enter amount of commands:\n", 26) = 26
read(0, "1\n", 512)                = 2
write(4, "\2\0\0\0", 4)             = 4
write(4, "\0\0\0\0", 4)             = 4
read(0, "1 2 3\n", 512)            = 6

```

```

write(4, "\3\0\0\0", 4)      = 4
write(4, "\1\0\0\0", 4)      = 4
write(4, "\2\0\0\0", 4)      = 4
write(4, "\3\0\0\0", 4)      = 4
close(4)                      = 0
close(3)                      = 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

clone(...) - создает новый процесс

pipe(...) - системный вызов возвращает пару дескрипторов, 3 и 4. Цифры таковы, потому что никаких дескрипторов, кроме 0 (stdin), 1 (stdout) и 2 (stderr), пока не было выделено процессу.

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.