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: Прохоров Данила Михайлович

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**

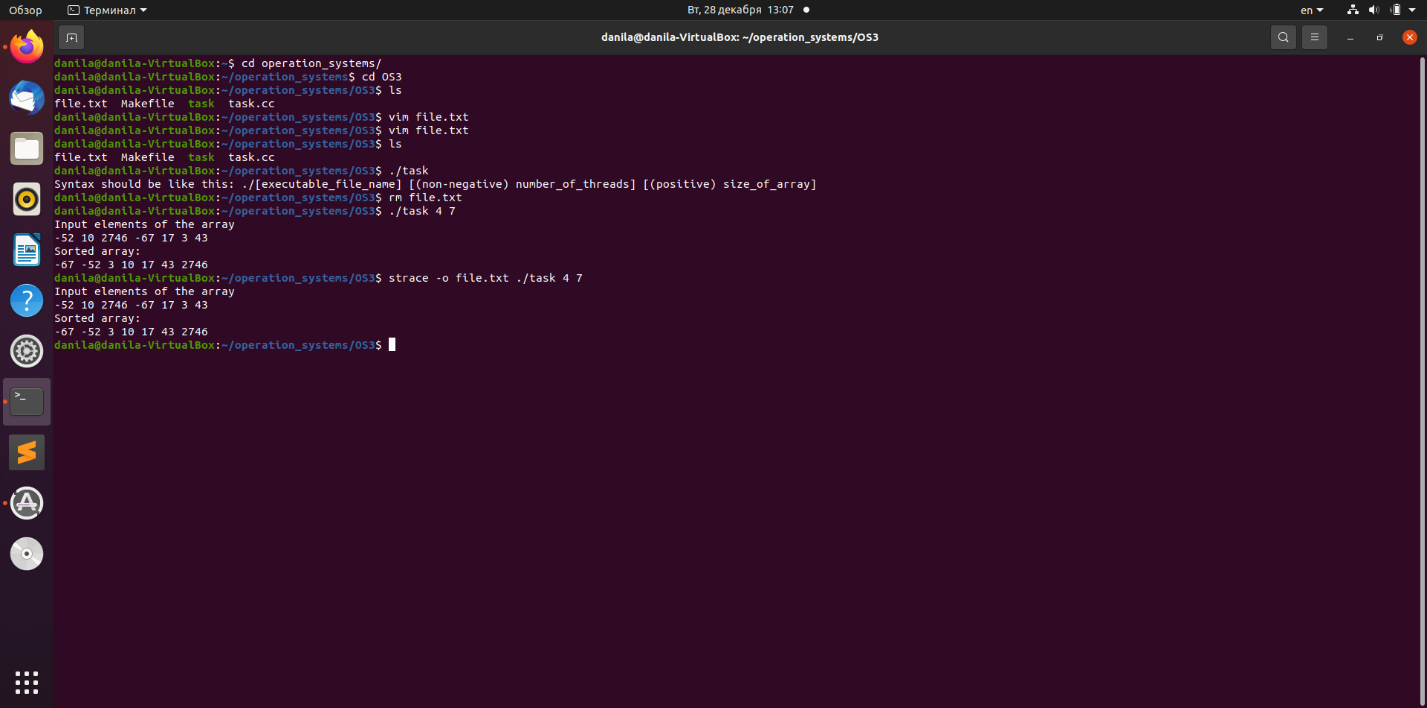
https://github.com/dmprokhorov

**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**

Запуск программы:



Исходный код strace:

execve("./task", ["./task", "4", "7"], 0x7ffdbe439900 /\* 49 vars \*/) = 0

|  |
| --- |
|  |
|  | brk(NULL) = 0x5614331b7000 |
|  | arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fffe6e665b0) = -1 EINVAL (Недопустимый аргумент) |
|  | 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 |
|  | fstat(3, {st\_mode=S\_IFREG|0644, st\_size=74405, ...}) = 0 |
|  | mmap(NULL, 74405, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f698880e000 |
|  | 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) = 0x7f698880c000 |
|  | 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) = 0x7f69887e9000 |
|  | mmap(0x7f69887f0000, 69632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7000) = 0x7f69887f0000 |
|  | mmap(0x7f6988801000, 20480, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f6988801000 |
|  | mmap(0x7f6988806000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c000) = 0x7f6988806000 |
|  | mmap(0x7f6988808000, 13432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f6988808000 |
|  | 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) = 0x7f69885f7000 |
|  | mprotect(0x7f698861c000, 1847296, PROT\_NONE) = 0 |
|  | mmap(0x7f698861c000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7f698861c000 |
|  | mmap(0x7f6988794000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7f6988794000 |
|  | mmap(0x7f69887df000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f69887df000 |
|  | mmap(0x7f69887e5000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f69887e5000 |
|  | close(3) = 0 |
|  | mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f69885f4000 |
|  | arch\_prctl(ARCH\_SET\_FS, 0x7f69885f4740) = 0 |
|  | mprotect(0x7f69887df000, 12288, PROT\_READ) = 0 |
|  | mprotect(0x7f6988806000, 4096, PROT\_READ) = 0 |
|  | mprotect(0x561432234000, 4096, PROT\_READ) = 0 |
|  | mprotect(0x7f698884e000, 4096, PROT\_READ) = 0 |
|  | munmap(0x7f698880e000, 74405) = 0 |
|  | set\_tid\_address(0x7f69885f4a10) = 3846 |
|  | set\_robust\_list(0x7f69885f4a20, 24) = 0 |
|  | rt\_sigaction(SIGRTMIN, {sa\_handler=0x7f69887f0bf0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7f69887fe3c0}, NULL, 8) = 0 |
|  | rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f69887f0c90, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f69887fe3c0}, NULL, 8) = 0 |
|  | rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0 |
|  | prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0 |
|  | fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0 |
|  | brk(NULL) = 0x5614331b7000 |
|  | brk(0x5614331d8000) = 0x5614331d8000 |
|  | write(1, "Input elements of the array\n", 28) = 28 |
|  | fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0 |
|  | read(0, "-52 10 2746 -67 17 3 43\n", 1024) = 24 |
|  | mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f6987df3000 |
|  | mprotect(0x7f6987df4000, 8388608, PROT\_READ|PROT\_WRITE) = 0 |
|  | clone(child\_stack=0x7f69885f2fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[3850], tls=0x7f69885f3700, child\_tidptr=0x7f69885f39d0) = 3850 |
|  | futex(0x7f69885f39d0, FUTEX\_WAIT, 3850, NULL) = 0 |
|  | write(1, "Sorted array:\n", 14) = 14 |
|  | write(1, "-67 -52 3 10 17 43 2746 \n", 25) = 25 |
|  | lseek(0, -1, SEEK\_CUR) = -1 ESPIPE (Недопустимая операция смещения) |
|  | 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(…) – создаёт дочерний процесс

futex(…) – ожидание, пока не произойдёт какое-то событие.

set\_tid\_address(…) – при запуске потока с помощью clone с флагом CLONE\_CHILD\_SETTID в значение set\_child\_tid устанавливается равным аргументу системного вызова ctid

set\_robust\_list(…) – возвращает начало списка надёжных фьютексов нити (потока)

rt\_sigaction(…) - изменениt выполняемого процессом действия при получении определённого сигнала

rt\_sigprocmask(…) - выборка и/или изменение маски сигналов вызывающей нити

prlimit64(…) – устанавливает ограничение ресурса для процесса

pread64(…) – читает из файлового дескриптора

getcwd(…) - получить текущую рабочую директорию

**Выводы**В данной лабораторной работе я углублённо разобрал strace и убедился, что это очень полезная утилита для просмотра системных вызовов.