

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: Меджидли Исмаил Ибрагим оглы  
Группа: М8О-201Б-21  
Преподаватель: Миронов Евгений Сергеевич  
Оценка: \_\_\_\_\_  
Дата: \_\_\_\_\_  
Подпись: \_\_\_\_\_

Москва, 2023

## **Содержание**

1. Репозиторий
2. Постановка задачи
3. Общий метод и алгоритм решения
4. Демонстрация работы
5. Выводы

## **Репозиторий**

<https://github.com/imedzhidli/Operational-Systems>

## **Постановка задачи**

Продемонстрировать работу утилиты *strace* на примере лабораторной работы

## **Общие сведения о программе**

*strace* — это утилита, отслеживающая системные вызовы, которые представляют собой механизм трансляции, обеспечивающий интерфейс между процессом и операционной системой (ядром). Эти вызовы могут быть перехвачены и прочитаны. Это позволяет лучше понять, что процесс пытается сделать в заданное время. Перехватывая эти вызовы, мы можем добиться лучшего понимания поведения процессов, особенно если что-то идет не так.

## **Общий метод и алгоритм решения**

Я продемонстрирую работу утилиты *strace* на примере 2 лабораторной работы.

В этой работе я использовал следующие системные вызовы: `pipe`, `fork`, `dup2`, `exec1`.

*Strace* показал все системные вызовы, а также вызовы чтения и записи (вызовы `read/write` число прочитанных/записанных байт).

Важно: `fork` вызывает `clone` для создания дочернего процесса.

## Демонстрация работы

```
imedzhidli@imedzhidli:~/Desktop/OS/2Laba/build$ strace ./lab2 < test.txt
execve("./lab2", ["/lab2"], 0x7fffa6e803b0 /* 108 vars */) = 0
brk(NULL)                                = 0x55722450f000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffdcfc34cf0) = -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=75706, ...}) = 0
mmap(NULL, 75706, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fd27ba63000
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) = 0x7fd27ba61000
mmap(NULL, 1972224, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7fd27b87f000
mprotect(0x7fd27b915000, 1290240, PROT_NONE) = 0
mmap(0x7fd27b915000, 987136, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x96000) =
0x7fd27b915000
mmap(0x7fd27ba06000, 299008, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x187000) =
0x7fd27ba06000
```

```

mmap(0x7fd27ba50000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d0000) =
0x7fd27ba50000
mmap(0x7fd27ba5e000, 10240, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fd27ba5e000
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\3405\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) = 0x7fd27b864000
mmap(0x7fd27b867000, 73728, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) =
0x7fd27b867000
mmap(0x7fd27b879000, 16384, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x15000) =
0x7fd27b879000
mmap(0x7fd27b87d000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) =
0x7fd27b87d000
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\3\0>\0\1\0\0\0\300A\2\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"...,
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", 32,
848) = 32
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\fiQ\226\236i\253-'o"..., 68, 880) =
68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029592, ...}) = 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"...,
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", 32,
848) = 32
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\fiQ\226\236i\253-'o"..., 68, 880) =
68
mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7fd27b672000
mmap(0x7fd27b694000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) =
0x7fd27b694000
mmap(0x7fd27b80c000, 319488, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19a000) =
0x7fd27b80c000
mmap(0x7fd27b85a000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) =
0x7fd27b85a000
mmap(0x7fd27b860000, 13920, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fd27b860000
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\3\0>\0\1\0\0\0\300\323\0\0\0\0\0\0"...,
832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1369384, ...}) = 0
mmap(NULL, 1368336, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x7fd27b523000
mmap(0x7fd27b530000, 684032, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xd000) =
0x7fd27b530000
mmap(0x7fd27b5d7000, 626688, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xb4000) =
0x7fd27b5d7000
mmap(0x7fd27b670000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x14c000) =
0x7fd27b670000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fd27b521000
arch_prctl(ARCH_SET_FS, 0x7fd27b522100) = 0
mprotect(0x7fd27b85a000, 16384, PROT_READ) = 0
mprotect(0x7fd27b670000, 4096, PROT_READ) = 0
mprotect(0x7fd27b87d000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fd27b51f000
mprotect(0x7fd27ba50000, 45056, PROT_READ) = 0
mprotect(0x55722395f000, 4096, PROT_READ) = 0
mprotect(0x7fd27baa3000, 4096, PROT_READ) = 0
munmap(0x7fd27ba63000, 75706) = 0
brk(NULL) = 0x55722450f000
brk(0x557224530000) = 0x557224530000
fstat(0, {st_mode=S_IFREG|0664, st_size=45, ...}) = 0

```

```

read(0, "imedy.txt\n1 2 3 4 5\n100 2 5\n500 "..., 4096) = 45
pipe([3, 4]) = 0
clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fd27b5223d0) = 28652
close(3) = 0
write(4, "1 2 3 4 5\n", 10) = 10
write(4, "100 2 5\n", 8) = 8
write(4, "500 2 6 6\n", 10) = 10
write(4, "10 2 0\n", 7) = 7
read(0, "", 4096) = 0
close(4) = 0
wait4(-1, Division by zero.
NULL, 0, NULL) = 28652
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=28652,
si_uid=1000, si_status=1, si_utime=0, si_stime=0} ---
exit_group(0) = ?
+++ exited with 0 +++
imedzhidli@imedzhidli:~/Desktop/OS/2Laba/build$

```

## Выводы:

Во время выполнения работы я изучил утилиту strace, понял, что она может быть полезной во время поиска ошибок.