# Министерство науки и высшего образования РФ Федеральное государственное бюджетное образовательное учреждение

**высшего образования**

# «Московский Авиационный Институт» Национальный Исследовательский Университет

**Институт** №8 «Информационные технологии и прикладная математика»

**Кафедра** 806 «Вычислительная математика и программирование»

# Лабораторная работа №1

**по курсу «Операционные системы»**

|  |  |
| --- | --- |
| Студент: | Гуликов К.А. |
| Группа: | М8О-206Б-20 |
| Преподаватель: | Миронов Е. С. |
| Подпись: |  |
| Оценка: |  |
| Дата: |  |

Москва, 2022

# Содержание

1. Постановка задачи;
2. Общие сведения о Strace;
3. Strace;
4. Вывод.

# Постановка задачи

При выполнении лабораторных работ необходимо продемонстрировать ключевые системные вызовы, которые в них используются.

# Общие сведения о Strace

Strace - это утилита, которая отслеживает системные вызовы, которые представляют собой механизм трансляции, обеспечивающий интерфейс между процессом и операционной системой. Эти вызовы могут быть перехвачены и прочитаны. Это позволяет лучше понять, что процесс пытается сделать в заданное время. Перехватывая эти вызовы, можно добиться лучшего понимания поведения процессов, особенно если что-то идет не так.

# Strace

**Lab 2**

konstantin@LAPTOP-44CRFC1U:~/labs/os/lab2$ strace ./ga1 < test.txt execve("./ga1", ["./ga1"], 0x7fffe198ad70 /\* 19 vars \*/) = 0

brk(NULL) = 0x7fffc4bbd000

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=47603, ...}) = 0

mmap(NULL, 47603, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f60228f5000

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\0\3\0>\0\1\0\0\0\260\34\2\0\0\0\0\0"..., 832) = 832 fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2030544, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f60228f0000

mmap(NULL, 4131552, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f6022200000

mprotect(0x7f60223e7000, 2097152, PROT\_NONE) = 0

mmap(0x7f60225e7000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f60225e7000 mmap(0x7f60225ed000, 15072, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f60225ed000

close(3) = 0

arch\_prctl(ARCH\_SET\_FS, 0x7f60228f14c0) = 0 mprotect(0x7f60225e7000, 16384, PROT\_READ) = 0

mprotect(0x7f6022c01000, 4096, PROT\_READ) = 0

mprotect(0x7f6022827000, 4096, PROT\_READ) = 0

munmap(0x7f60228f5000, 47603) = 0

fstat(1, {st\_mode=S\_IFCHR|0660, st\_rdev=makedev(4, 1), ...}) = 0 ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0 brk(NULL) = 0x7fffc4bbd000 brk(0x7fffc4bde000) = 0x7fffc4bde000

pipe([3, 4]) = 0

## clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD,

**child\_tidptr=0x7f60228f1790) = 109**

close(4) = 0

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

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=109, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

write(1, "Enter file name to read data: 4\n", 32Enter file name to read data: 4

) = 32

read(3, "\5\0\0\0", 4) = 4

write(1, "5\n", 25

) = 2

read(3, "\350\377\377\377", 4) = 4

write(1, "-24\n", 4-24

) = 4

read(3, "\6\0\0\0", 4) = 4

write(1, "6\n", 26

) = 2

read(3, "", 4) = 0

close(3) = 0

exit\_group(0) = ?

+++ exited with 0 +++

# Вывод

Такие утилиты, как strace или, например, ltrace являются простыми и довольно надежными инструментами, которые позволяют следить за процессом работы программы, отслеживая используемые системные вызовы. Анализируя вывод strace можно узнать как именно работает программа, как совершает свои вызовы, в каком порядке, а эта информация крайне полезна при дебаггинге или оптимизации. С помощью различных ключей можно применять различные фильтры для более удобного поиска необходимой информации.

В целом, это очень удобный инструмент, который поможет изучить принципы работы программы и отследить возникшие ошибки.