Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №1 по курсу**

**«Операционные системы»**

**Средства диагностики в ОС**

Студент: Е. А. Айрапетова

Группа: М8О–206Б–20

Вариант: 16

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2020.

**Постановка задачи**

## Цель работы

Приобретение практических навыков диагностики работы программного обеспечения.

## Задание

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

**Описание используемых утилит**

Strace — это утилита Linux, отслеживающая системные вызовы, которые представляют собой механизм трансляции, обеспечивающий интерфейс между процессом и операционной системой. Использование данной утилиты позволяет понять, что процесс пытается сделать в данное время. Strace может быть полезен при отладке программ.

Для удобства работы с протоколом утилиты можно использовать следующие ключи:

-o file – Перенаправить протокол утилиты в файл file.

-e trace=filters – Указать выражения, по которым будут фильтроваться системные вызовы. Например -e trace=write,%process задаёт фильтрацию по системным вызовам write и по группе системных вызовов, связанных с межпроцессорным взаимодействием.

-f – Отслеживать системные вызовы в дочерних процессах.

-y – Заменить в протоколе все файловые дескрипторы на имена соответствующих им файлов (где возможно).

-p file – Отслеживать только обращения к файлу file.

-k – Отображать стек вызовов.

**Пример использования**

В качестве основы будем использовать код лабораторной №2. Далее представлен результат работы лабораторной работы при ее вызове с помощью strace -f ./a.out < test:

jane@Evgenia:/mnt/c/Files/ОСи/ЛР1$ strace -f ./a.out < test > output.txt

execve("./a.out", ["./a.out"], 0x7fffde110a38 /\* 19 vars \*/) = 0

brk(NULL) = 0x7fffe0ff3000

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=33854, ...}) = 0

mmap(NULL, 33854, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fdc7fd48000

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

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2030928, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdc7fd40000

mmap(NULL, 4131552, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdc7f600000

mprotect(0x7fdc7f7e7000, 2097152, PROT\_NONE) = 0

mmap(0x7fdc7f9e7000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fdc7f9e7000

mmap(0x7fdc7f9ed000, 15072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fdc7f9ed000

close(3) = 0

arch\_prctl(ARCH\_SET\_FS, 0x7fdc7fd414c0) = 0

mprotect(0x7fdc7f9e7000, 16384, PROT\_READ) = 0

mprotect(0x7fdc80001000, 4096, PROT\_READ) = 0

mprotect(0x7fdc7fc29000, 4096, PROT\_READ) = 0

munmap(0x7fdc7fd48000, 33854) = 0

fstat(0, {st\_mode=S\_IFREG|0777, st\_size=97, ...}) = 0

brk(NULL) = 0x7fffe0ff3000

brk(0x7fffe1014000) = 0x7fffe1014000

read(0, "output.txt\nRigth string\nThis is "..., 4096) = 97

openat(AT\_FDCWD, "output.txt", O\_WRONLY|O\_CREAT|O\_APPEND, 0600) = 3

pipe([4, 5]) = 0

pipe([6, 7]) = 0

clone(strace: Process 3663 attached

child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fdc7fd41790) = 3663

[pid 3663] close(5 <unfinished ...>

[pid 3662] close(4) = 0

[pid 3662] close(7 <unfinished ...>

[pid 3663] <... close resumed> ) = 0

[pid 3662] <... close resumed> ) = 0

[pid 3663] close(6 <unfinished ...>

[pid 3662] write(5, "\r\0\0\0", 4 <unfinished ...>

[pid 3663] <... close resumed> ) = 0

[pid 3662] <... write resumed> ) = 4

[pid 3662] write(5, "Rigth string\0", 13 <unfinished ...>

[pid 3663] read(4, <unfinished ...>

[pid 3662] <... write resumed> ) = 13

[pid 3662] read(6, <unfinished ...>

[pid 3663] <... read resumed> "\r\0\0\0", 4) = 4

[pid 3663] read(4, "Rigth string\0", 13) = 13

[pid 3663] write(7, "\1", 1 <unfinished ...>

[pid 3662] <... read resumed> "\1", 1) = 1

[pid 3662] write(5, "\17\0\0\0", 4 <unfinished ...>

[pid 3663] <... write resumed> ) = 1

[pid 3662] <... write resumed> ) = 4

[pid 3662] write(5, "This is string\0", 15 <unfinished ...>

[pid 3663] write(3, "Rigth string\0", 13 <unfinished ...>

[pid 3662] <... write resumed> ) = 15

[pid 3663] <... write resumed> ) = 13

[pid 3662] read(6, <unfinished ...>

[pid 3663] write(3, "\n", 1) = 1

[pid 3663] read(4, "\17\0\0\0", 4) = 4

[pid 3663] read(4, "This is string\0", 15) = 15

[pid 3663] write(7, "\1", 1) = 1

[pid 3662] <... read resumed> "\1", 1) = 1

[pid 3662] write(5, "\r\0\0\0", 4 <unfinished ...>

[pid 3663] write(3, "This is string\0", 15 <unfinished ...>

[pid 3662] <... write resumed> ) = 4

[pid 3663] <... write resumed> ) = 15

[pid 3662] write(5, "wrong string\0", 13 <unfinished ...>

[pid 3663] write(3, "\n", 1 <unfinished ...>

[pid 3662] <... write resumed> ) = 13

[pid 3662] read(6, <unfinished ...>

[pid 3663] <... write resumed> ) = 1

[pid 3663] read(4, "\r\0\0\0", 4) = 4

[pid 3663] read(4, "wrong string\0", 13) = 13

[pid 3663] write(7, "\0", 1) = 1

[pid 3662] <... read resumed> "\0", 1) = 1

[pid 3663] read(4, <unfinished ...>

[pid 3662] write(3, "ERROR\n", 6) = 6

[pid 3662] write(5, "\t\0\0\0", 4) = 4

[pid 3663] <... read resumed> "\t\0\0\0", 4) = 4

[pid 3662] write(5, "13232142\0", 9 <unfinished ...>

[pid 3663] read(4, <unfinished ...>

[pid 3662] <... write resumed> ) = 9

[pid 3662] read(6, <unfinished ...>

[pid 3663] <... read resumed> "13232142\0", 9) = 9

[pid 3663] write(7, "\0", 1 <unfinished ...>

[pid 3662] <... read resumed> "\0", 1) = 1

[pid 3663] <... write resumed> ) = 1

[pid 3662] write(3, "ERROR\n", 6 <unfinished ...>

[pid 3663] read(4, <unfinished ...>

[pid 3662] <... write resumed> ) = 6

[pid 3662] write(5, "\n\0\0\0", 4) = 4

[pid 3662] write(5, "Eeeeeeeee\0", 10 <unfinished ...>

[pid 3663] <... read resumed> "\n\0\0\0", 4) = 4

[pid 3662] <... write resumed> ) = 10

[pid 3662] read(6, <unfinished ...>

[pid 3663] read(4, "Eeeeeeeee\0", 10) = 10

[pid 3663] write(7, "\1", 1) = 1

[pid 3662] <... read resumed> "\1", 1) = 1

[pid 3663] write(3, "Eeeeeeeee\0", 10 <unfinished ...>

[pid 3662] write(5, "\r\0\0\0", 4 <unfinished ...>

[pid 3663] <... write resumed> ) = 10

[pid 3662] <... write resumed> ) = 4

[pid 3662] write(5, "AAAAAAAAAAAA\0", 13 <unfinished ...>

[pid 3663] write(3, "\n", 1 <unfinished ...>

[pid 3662] <... write resumed> ) = 13

[pid 3662] read(6, <unfinished ...>

[pid 3663] <... write resumed> ) = 1

[pid 3663] read(4, "\r\0\0\0", 4) = 4

[pid 3663] read(4, "AAAAAAAAAAAA\0", 13) = 13

[pid 3663] write(7, "\1", 1) = 1

[pid 3662] <... read resumed> "\1", 1) = 1

[pid 3663] write(3, "AAAAAAAAAAAA\0", 13 <unfinished ...>

[pid 3662] write(5, "\f\0\0\0", 4) = 4

[pid 3663] <... write resumed> ) = 13

[pid 3662] write(5, "fffff123FFF\0", 12 <unfinished ...>

[pid 3663] write(3, "\n", 1 <unfinished ...>

[pid 3662] <... write resumed> ) = 12

[pid 3662] read(6, <unfinished ...>

[pid 3663] <... write resumed> ) = 1

[pid 3663] read(4, "\f\0\0\0", 4) = 4

[pid 3663] read(4, "fffff123FFF\0", 12) = 12

[pid 3663] write(7, "\0", 1 <unfinished ...>

[pid 3662] <... read resumed> "\0", 1) = 1

[pid 3662] write(3, "ERROR\n", 6 <unfinished ...>

[pid 3663] <... write resumed> ) = 1

[pid 3662] <... write resumed> ) = 6

[pid 3663] read(4, <unfinished ...>

[pid 3662] read(0, "", 4096) = 0

[pid 3662] write(5, "\0\0\0\0", 4) = 4

[pid 3662] wait4(-1, <unfinished ...>

[pid 3663] <... read resumed> "\0\0\0\0", 4) = 4

[pid 3663] close(3) = 0

[pid 3663] close(4) = 0

[pid 3663] close(7) = 0

[pid 3663] lseek(0, -86, SEEK\_CUR) = 11

[pid 3663] exit\_group(0) = ?

[pid 3663] +++ exited with 0 +++

<... wait4 resumed> NULL, 0, NULL) = 3663

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=3663, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

close(3) = 0

close(5) = 0

close(6) = 0

exit\_group(0) = ?

+++ exited with 0 +++

**Вывод**

Утилита strace – простой и надежный инструмент для отслеживания системных вызовов. Они полезны при отладке и тестировании программ на практике, так как при работе с системными вызовами может происходить множество ошибок. Несмотря на то, что, на первый взгляд, протоколы этих утилит кажутся чересчур объёмными и непонятными, при помощи различных ключей можно отфильтровать их и оставить только то, что нужно в конкретной ситуации.