

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Компьютерные науки и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Лабораторные работы №8
По курсу «Операционные системы»

Студент: Попов А. Д.

Группа: М8О-208Б-23

Преподаватель: Живалев Е. А.

Дата: _____

Оценка: _____

Подпись: _____

Москва, 2024

Описание и определение strace:

strace — это эффективный инструмент для диагностики и отладки, используемый в Unix-подобных операционных системах, таких как Linux. Он позволяет отслеживать и фиксировать системные вызовы, выполняемые процессом, а также сигналы, которые он обрабатывает. Системные вызовы представляют собой интерфейс взаимодействия между пользовательскими приложениями и ядром ОС. С помощью strace можно детально изучить, какие файлы открывает программа, какие сетевые соединения использует, какие ошибки возникают в процессе выполнения и многое другое. Этот инструмент особенно полезен для отладки, анализа производительности и изучения поведения процессов в реальном времени.

Репозиторий: https://github.com/alpopov/OS_labs/tree/master/LW8

Листинг работы strace:

```
execve("./build/LW3/lab3", ["./build/LW3/lab3"], 0x7ffffa5e5c60 /* 48 vars
*/) = 0
brk(NULL) = 0x5dfa34fbf000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x729412a4b000
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=59691, ...}) = 0
mmap(NULL, 59691, PROT_READ, MAP_PRIVATE, 3, 0) = 0x729412a3c000
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\0\0\3\0>\0\1\0\0\0\220\243\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\0\0\0\0"...
, 784, 64)
= 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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\0\0\0\0"...
, 784, 64)
= 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x729412800000
mmap(0x729412828000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x729412828000
mmap(0x7294129b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1b0000) = 0x7294129b0000
mmap(0x7294129ff000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7294129ff000
mmap(0x729412a05000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x729412a05000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x729412a39000
arch_prctl(ARCH_SET_FS, 0x729412a39740) = 0
set_tid_address(0x729412a39a10) = 6478
set_robust_list(0x729412a39a20, 24) = 0
rseq(0x729412a3a060, 0x20, 0, 0x53053053) = 0
mprotect(0x7294129ff000, 16384, PROT_READ) = 0
```

[illegible]

```

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=6481, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---
munmap(0x729412a49000, 32)          = 0
munmap(0x729412a4a000, 32)          = 0
unlink("/dev/shm/sem.semaphore_write") = 0
unlink("/dev/shm/sem.semaphore_read") = 0
close(3)                          = 0
unlink("/dev/shm/virtual_memory")   = 0
close(4)                          = 0
exit_group(0)                      = ?
+++ exited with 0 +++

```

Комментарии к листингу:

Если обобщить вывод, то он разделён на 3 последовательных этапа.

Сначала с помощью различных системных вызовов инициализируется программа, загружаются нужные библиотеки, устанавливаются соответствующие значения.

Затем программа начинает выполнение. Происходит вызов кода из программы — считывается название некоторого файла. Затем идёт создание семафоров, открытие файла, создание дочернего процесса и вывод значений

Последним этапом программа завершает работу: родительский процесс ждёт завершения дочернего, удаляются все семафоры, разделяемая память, освобождается отображенная память и программа выходит.

Вывод:

В рамках данной лабораторной работы инструмент strace был применен для детального анализа выполнения программы. Были изучены все системные вызовы, их параметры и возвращаемые значения. В процессе работы удалось глубже понять логику программы, а также убедиться в отсутствии критических ошибок, что свидетельствует о её стабильной и корректной работе.