## 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"

Лабораторная работа №8 По курсу «Операционные системы»

C	тудент: Попов А. Д.
Гр	уппа: М8О-208Б-23
Преподава	тель: Миронов Е. С.
Дата:	
Оценка:	
П	
Подпись:	

#### Описание и определение strace:

strace — это мощный диагностический и отладочный инструмент в Unixподобных операционных системах, включая Linux. Он позволяет отслеживать и записывать системные вызовы, которые выполняет процесс, а также получает. Системные сигналы, которые ОН вызовы взаимодействия между пользовательским программным обеспечением и ядром операционной системы. Используя strace, можно получить подробную информацию о том, какие файлы открывает программа, какие сетевые соединения устанавливает, какие ошибки возникают при выполнении и многое другое. Это особенно полезно для отладки программ, диагностики проблем с производительностью и анализа поведения процессов в реальном времени.

#### Листинг работы strace:

```
execve("./build/LW3/lab3", ["./build/LW3/lab3"], 0x7ffffa5e5c60 /* 48 vars */) = 0
                       = 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
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY/O_CLOEXEC) = 3
fstat(3, \{st\_mode = S\_IFREG/0755, st\_size = 2125328, ...\}) = 0
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) = 0
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,
-1.00 = 0x729412a05000
close(3)
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)
set_robust_list(0x729412a39a20, 24)
rseq(0x729412a3a060, 0x20, 0, 0x53053053) = 0
mprotect(0x7294129ff000, 16384, PROT\_READ) = 0
mprotect(0x5dfa34281000, 4096, PROT\_READ) = 0
mprotect(0x729412a83000, 8192, PROT READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x729412a3c000, 59691)
                                = 0
getrandom("\x39\x56\xf4\x4c\x41\x85\xee\x97", 8, GRND\_NONBLOCK) = 8
brk(NULL)
                       = 0x5dfa34fbf000
brk(0x5dfa34fe0000)
                          = 0x5dfa34fe0000
fstat(0, \{st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...\}) = 0
read(0, "a\n", 1024)
unlink("/dev/shm/sem.semaphore write") = 0
unlink("/dev/shm/sem.semaphore\_read") = 0
```

```
openat(AT FDCWD, "/dev/shm/sem.semaphore write", O RDWR/O NOFOLLOW/O CLOEXEC) = -1 ENOENT
(No such file or directory)
getrandom("\x3a\x59\xc4\xd3\xcb\xb4\x1d\xc7", 8, GRND NONBLOCK) = 8
newfstatat(AT_FDCWD, "/dev/shm/sem.wWTIG7", 0x7fff4f4c1640, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (No
                                                          "/dev/shm/sem.wWTIG7", O RDWR/O CREAT/O EXCL/O NOFOLLOW/O CLOEXEC,
openat(AT_FDCWD,
0666) = 3
write(3, "\langle 0 \rangle 0 \rangle 0 \langle 0 \rangle 
mmap(NULL, 32, PROT\_READ/PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x729412a4a000
link("/dev/shm/sem.wWTIG7", "/dev/shm/sem.semaphore_write") = 0
fstat(3, \{st\_mode=S\_IFREG/0664, st\_size=32, ...\}) = 0
unlink("/dev/shm/sem.wWTIG7")
                                                                                                = 0
close(3)
openat(AT_FDCWD, "/dev/shm/sem.semaphore_read", O_RDWR/O_NOFOLLOW/O_CLOEXEC) = -1 ENOENT
(No such file or directory)
getrandom("\xf9\xa8\x98\x25\x1c\x68\x8f\x59", 8, GRND\_NONBLOCK) = 8
newfstatat(AT_FDCWD, "/dev/shm/sem.7rScGO", 0x7fff4f4c1640, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (No
such file or directory)
                                                           "/dev/shm/sem.7rScGO", O RDWR/O CREAT/O EXCL/O NOFOLLOW/O CLOEXEC,
openat(AT FDCWD,
0666) = 3
mmap(NULL, 32, PROT\_READ/PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x729412a49000
link("/dev/shm/sem.7rScGO", "/dev/shm/sem.semaphore_read") = 0
fstat(3, \{st \ mode=S \ IFREG/0664, st \ size=32, ...\}) = 0
unlink("/dev/shm/sem.7rScGO")
close(3)
                                                                    = 0
openat(AT_FDCWD, "/dev/shm/virtual_memory", O_RDWR/O_CREAT/O_NOFOLLOW/O_CLOEXEC, 0666) = 3
ftruncate(3, 10240)
                                                                              = 0
openat(AT_FDCWD, "a", O_RDONLY)
                                                                                                              = 4
                                                                                                    flags=CLONE_CHILD_CLEARTID/CLONE_CHILD_SETTID/SIGCHLD,
clone(child_stack=NULL,
child\_tidptr=0x729412a39a10) = 6481
mmap(NULL, 10240, PROT_READ/PROT_WRITE, MAP_SHARED, 3, 0) = 0x729412a46000
                                                                               FUTEX_WAIT_BITSET/FUTEX_CLOCK_REALTIME,
                                                                                                                                                                                                                                              0,
futex(0x729412a49000,
                                                                                                                                                                                                                                                                          NULL.
FUTEX\_BITSET\_MATCH\_ANY) = 0
fstat(1, \{st\_mode=S\_IFCHR/0620, st\_rdev=makedev(0x88, 0), ...\}) = 0
write(1, "26.80\n", 6)
                                                                             = 6
futex(0x729412a4a000, FUTEX_WAKE, 1) = 1
futex(0x729412a49000,
                                                                               FUTEX_WAIT_BITSET/FUTEX_CLOCK_REALTIME,
                                                                                                                                                                                                                                             0.
                                                                                                                                                                                                                                                                          NULL.
FUTEX\_BITSET\_MATCH\_ANY) = 0
write(1, "0.00\n", 5)
futex(0x729412a4a000, FUTEX_WAKE, 1) = 1
write(1, "7.51\n", 5)
                                                                             = 5
futex(0x729412a4a000, FUTEX_WAKE, 1) = 1
munmap(0x729412a46000, 1024)
                                                                                                   = 0
wait4(6481, [\{WIFEXITED(s) \&\& WEXITSTATUS(s) == 0\}], 0, NULL) = 6481
--- 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")
close(4)
                                                                   = 0
exit_group(0)
                                                                         = ?
+++ exited with 0+++
```

### Комментарии к листингу:

Если обобщить вывод, то он разделён на 3 блока: запуск, работа и завершение работы.

- 1. Сначала с помощью различных системных вызовов инициализируется программа, загружаются нужные библиотеки, устанавливаются соответствующие значения.
- 2. 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
- 3. Последним этапом программа завершает работу: родительский процесс ждёт завершения дочернего, удаляются все семафоры, разделяемая память, освобождается отображенная память и программа выходит.

#### Вывод:

В данной лабораторной работе **strace** был использован для анализа работы программы. Были проанализированы все системные вызовы, их аргументы и возвращаемые значения. В ходе выполнения работы было улучшено понимание программы, а также не было обнаружено критических ошибок, что подтверждает стабильность работы программы.