

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Информационные технологий и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №1
по курсу "Операционные системы"
3 семестр

Студент: Леухин М. В.
Группа: М8О-206Б-20
Преподаватель: Соколов А. А.
Дата: 27.11.21
Оценка: 5
Подпись: _____

Москва, 2021

Содержание

1	Постановка задачи	3
2	Основная часть	4
2.1	Описание утилиты <code>strace</code>	4
2.2	Пример использования <code>strace</code>	4
3	Вывод	8

1 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 и то, что их использование соответствует варианту ЛР. По итогам выполнения всех лабораторных работ отчёт по данной ЛР должен содержать краткую сводку по исследованию последующих ЛР.

2 Основная часть

2.1 Описание утилиты strace

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.

2.2 Пример использования strace

Действие утилиты продемонстрировано на примере лабораторной работы №3

```
1  execve("./a.out", ["./a.out"], 0x7ffd68351f40 /* 67 vars */) = 0
2  brk(NULL)                                = 0x5572916a2000
3  arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe6aac7740) = -1 EINVAL (Invalid
   argument)
4  access("/etc/ld.so.preload", R_OK)        = -1 ENOENT (No such file or
   directory)
5  openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
6  fstat(3, {st_mode=S_IFREG|0644, st_size=87249, ...}) = 0
7  mmap(NULL, 87249, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f32c7b1f000
8  close(3)                                  = 0
9  openat(AT_FDCWD, "/lib/x86_64-linux-gnu/librt.so.1",
   O_RDONLY|O_CLOEXEC) = 3
10 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0
   7\0\0\0\0\0\0"..., 832) = 832
11 fstat(3, {st_mode=S_IFREG|0644, st_size=40040, ...}) = 0
12 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
   0) = 0x7f32c7b1d000
13 mmap(NULL, 44000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
   0x7f32c7b12000
14 mprotect(0x7f32c7b15000, 24576, PROT_NONE) = 0
15 mmap(0x7f32c7b15000, 16384, PROT_READ|PROT_EXEC,
   MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f32c7b15000
16 mmap(0x7f32c7b19000, 4096, PROT_READ,
   MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f32c7b19000
17 mmap(0x7f32c7b1b000, 8192, PROT_READ|PROT_WRITE,
   MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8000) = 0x7f32c7b1b000
18 close(3)                                  = 0
19 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0",
   O_RDONLY|O_CLOEXEC) = 3
20 read(3,
   "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\220\201\0\0\0\0\0"...,
   832) = 832
21 pread64(3,
   "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"...
   68, 824) = 68
22 fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0
```

```

23 pread64(3,
    "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"...
    68, 824) = 68
24 mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
    0x7f32c7aef000
25 mmap(0x7f32c7af6000, 69632, PROT_READ|PROT_EXEC,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f32c7af6000
26 mmap(0x7f32c7b07000, 20480, PROT_READ,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7f32c7b07000
27 mmap(0x7f32c7b0c000, 8192, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000) = 0x7f32c7b0c000
28 mmap(0x7f32c7b0e000, 13432, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f32c7b0e000
29 close(3) = 0
30 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6",
    O_RDONLY|O_CLOEXEC) = 3
31 read(3,
    "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"... ,
    832) = 832
32 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"... ,
    784, 64) = 784
33 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
34 pread64(3,
    "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276
    68, 880) = 68
35 fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
36 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"... ,
    784, 64) = 784
37 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
38 pread64(3,
    "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276
    68, 880) = 68
39 mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
    0x7f32c78fd000
40 mprotect(0x7f32c7922000, 1847296, PROT_NONE) = 0
41 mmap(0x7f32c7922000, 1540096, PROT_READ|PROT_EXEC,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f32c7922000
42 mmap(0x7f32c7a9a000, 303104, PROT_READ,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f32c7a9a000
43 mmap(0x7f32c7ae5000, 24576, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f32c7ae5000
44 mmap(0x7f32c7aeb000, 13528, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f32c7aeb000
45 close(3) = 0
46 mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
    0) = 0x7f32c78fa000
47 arch_prctl(ARCH_SET_FS, 0x7f32c78fa740) = 0

```

```

48 mprotect(0x7f32c7ae5000 , 12288, PROT_READ) = 0
49 mprotect(0x7f32c7b0c000 , 4096, PROT_READ) = 0
50 mprotect(0x7f32c7b1b000 , 4096, PROT_READ) = 0
51 mprotect(0x557290478000 , 4096, PROT_READ) = 0
52 mprotect(0x7f32c7b62000 , 4096, PROT_READ) = 0
53 munmap(0x7f32c7b1f000 , 87249) = 0
54 set_tid_address(0x7f32c78faa10) = 14370
55 set_robust_list(0x7f32c78faa20 , 24) = 0
56 rt_sigaction(SIGRTMIN, {sa_handler=0x7f32c7af6bf0 , sa_mask=[],
    sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f32c7b043c0}, NULL,
    8) = 0
57 rt_sigaction(SIGRT_1, {sa_handler=0x7f32c7af6c90 , sa_mask=[],
    sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO,
    sa_restorer=0x7f32c7b043c0}, NULL, 8) = 0
58 rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
59 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
    rlim_max=RLIM64_INFINITY}) = 0
60 statfs("/dev/shm/", {f_type=TMPFS_MAGIC, f_bsize=4096,
    f_blocks=761005, f_bfree=749279, f_bavail=749279, f_files=761005,
    f_ffree=760858, f_fsid={val=[0, 0]}, f_namelen=255, f_frsize=4096,
    f_flags=ST_VALID|ST_NOSUID|ST_NODEV}) = 0
61 futex(0x7f32c7b11390 , FUTEX_WAKE_PRIVATE, 2147483647) = 0
62 unlink("/dev/shm/sem.source_sem") = -1 ENOENT (No such file or
    directory)
63 unlink("/dev/shm/sem.response_sem") = -1 ENOENT (No such file or
    directory)
64 unlink("/dev/shm/source_shm") = -1 ENOENT (No such file or
    directory)
65 unlink("/dev/shm/response_shm") = -1 ENOENT (No such file or
    directory)
66 openat(AT_FDCWD, "/dev/shm/source_shm",
    O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0644) = 3
67 openat(AT_FDCWD, "/dev/shm/response_shm",
    O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0644) = 4
68 ftruncate(3, 4096) = 0
69 ftruncate(4, 4096) = 0
70 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) =
    0x7f32c7b61000
71 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) =
    0x7f32c7b34000
72 openat(AT_FDCWD, "/dev/shm/sem.source_sem", O_RDWR|O_NOFOLLOW) = -1
    ENOENT (No such file or directory)
73 getpid() = 14370
74 lstat("/dev/shm/Y1LNKN", 0x7ffe6aac7260) = -1 ENOENT (No such file or
    directory)
75 openat(AT_FDCWD, "/dev/shm/Y1LNKN", O_RDWR|O_CREAT|O_EXCL, 0644) = 5
76 write(5,
    "\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\0\0\0\0",
    32) = 32
77 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0) = 0x7f32c7b33000
78 link("/dev/shm/Y1LNKN", "/dev/shm/sem.source_sem") = 0
79 fstat(5, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0
80 brk(NULL) = 0x5572916a2000

```

```

81 brk(0x5572916c3000) = 0x5572916c3000
82 unlink("/dev/shm/YlLNKN") = 0
83 close(5) = 0
84 openat(AT_FDCWD, "/dev/shm/sem.response_sem", O_RDWR|O_NOFOLLOW) = -1
ENOENT (No such file or directory)
85 getpid() = 14370
86 lstat("/dev/shm/hJfDGO", 0x7ffe6aac7250) = -1 ENOENT (No such file or
directory)
87 openat(AT_FDCWD, "/dev/shm/hJfDGO", O_RDWR|O_CREAT|O_EXCL, 0644) = 5
88 write(5,
"\0\0\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0",
32) = 32
89 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0) = 0x7f32c7b32000
90 link("/dev/shm/hJfDGO", "/dev/shm/sem.response_sem") = 0
91 fstat(5, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0
92 unlink("/dev/shm/hJfDGO") = 0
93 close(5) = 0
94 clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f32c78faa10) = 14371
95 getpid() = 14370
96 fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
97 write(1, "[14370] Enter the name of file t"... , 41) = 41
98 read(0, "output.txt\n", 256) = 11
99 futex(0x7f32c7b33000, FUTEX_WAKE, 1) = 1
100 futex(0x7f32c7b32000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY) = 0
101 write(1, "[14370] Enter string: ", 22) = 22
102 read(0, "Valid string\n", 256) = 13
103 futex(0x7f32c7b33000, FUTEX_WAKE, 1) = 1
104 futex(0x7f32c7b32000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY) = -1 EAGAIN (Resource temporarily
unavailable)
105 write(1, "[14370] Enter string: ", 22) = 22
106 read(0, "error\n", 256) = 6
107 futex(0x7f32c7b33000, FUTEX_WAKE, 1) = 1
108 write(1, "Error: \"error\" is not valid.\n", 29) = 29
109 write(1, "[14370] Enter string: ", 22) = 22
110 read(0, "", 256) = 0
111 write(1, "\n", 1) = 1
112 close(3) = 0
113 close(4) = 0
114 unlink("/dev/shm/source_shm") = 0
115 unlink("/dev/shm/response_shm") = 0
116 munmap(0x7f32c7b61000, 4096) = 0
117 munmap(0x7f32c7b34000, 4096) = 0
118 unlink("/dev/shm/sem.source_sem") = 0
119 unlink("/dev/shm/sem.response_sem") = 0
120 exit_group(0) = ?
121 +++ exited with 0 +++

```

3 Вывод

Утилита `strace` является удобным инструментом для отслеживания того, какие системные вызовы использует программа, а также результат их выполнения, что крайне полезно в процессе отладки программы. Также использование данной утилиты позволяет увидеть то, что происходит при запуске и выполнении любой программы с точки зрения операционной системы.