

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»

**Приобретение практических навыков диагностики работы
программного обеспечения.**

Группа: М80-210Б-22

Студент: Бонокин Д.С.

Преподаватель: Соколов А.А.

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2023

Постановка задачи

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

Используемые утилиты: `strace`.

Листинг программы

`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

Примеры работы

Лабораторная работа 3 – взаимодействие между потоками

1. `execve("./main", ["/main", "qq"], 0x7ffc4e651e08 /* 68 vars */) = 0`

2. brk(NULL) = 0x5566b6305000
3. arch_prctl(0x3001 /* ARCH_??? */, 0x7fff4d0fb700) = -1 EINVAL (Invalid argument)
4. mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fd55c729000
5. access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
6. openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
7. newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=61711, ...}, AT_EMPTY_PATH) = 0
8. mmap(NULL, 61711, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fd55c719000
9. close(3) = 0
10. openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
11. read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
12. newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2260296, ...}, AT_EMPTY_PATH) = 0
13. mmap(NULL, 2275520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fd55c400000
14. mprotect(0x7fd55c49a000, 1576960, PROT_NONE) = 0
15. mmap(0x7fd55c49a000, 1118208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7fd55c49a000
16. mmap(0x7fd55c5ab000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ab000) = 0x7fd55c5ab000
17. mmap(0x7fd55c61b000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21a000) = 0x7fd55c61b000
18. mmap(0x7fd55c629000, 10432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fd55c629000
19. close(3) = 0
20. openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
21. read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
22. newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125488, ...}, AT_EMPTY_PATH) = 0
23. mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fd55c6f9000
24. mmap(0x7fd55c6fc000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fd55c6fc000
25. mmap(0x7fd55c713000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7fd55c713000
26. mmap(0x7fd55c717000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7fd55c717000
27. close(3) = 0
28. openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
29. read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"..., 832) = 832
30. 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
31. pread64(3, "\4\0\0\0\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0"..., 48, 848) = 48
32. pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0=\340\256\265?\356\25x\261\27\313A#\350"..., 68, 896) = 68
33. newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2216304, ...}, AT_EMPTY_PATH) = 0
34. 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
35. mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fd55c000000
36. mmap(0x7fd55c028000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fd55c028000
37. mmap(0x7fd55c1bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7fd55c1bd000

```

38. mmap(0x7fd55c215000, 24576, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7fd55c215000
39. mmap(0x7fd55c21b000, 52816, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fd55c21b000
40. close(3) = 0
41. openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
42. read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
43. newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
44. mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fd55c319000
45. mmap(0x7fd55c327000, 507904, PROT_READ|PROT_EXEC,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7fd55c327000
46. mmap(0x7fd55c3a3000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
    0x8a000) = 0x7fd55c3a3000
47. mmap(0x7fd55c3fe000, 8192, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7fd55c3fe000
48. close(3)
49. = 0
50. mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
    0x7fd55c6f7000
51. arch_prctl(ARCH_SET_FS, 0x7fd55c6f83c0) = 0
52. set_tid_address(0x7fd55c6f8690) = 12131
53. set_robust_list(0x7fd55c6f86a0, 24) = 0
54. rseq(0x7fd55c6f8d60, 0x20, 0, 0x53053053) = 0
55. mprotect(0x7fd55c215000, 16384, PROT_READ) = 0
56. mprotect(0x7fd55c3fe000, 4096, PROT_READ) = 0
57. mprotect(0x7fd55c717000, 4096, PROT_READ) = 0
58. mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
    0x7fd55c6f5000
59. mprotect(0x7fd55c61b000, 45056, PROT_READ) = 0
60. mprotect(0x5566b4b57000, 4096, PROT_READ) = 0
61. mprotect(0x7fd55c763000, 8192, PROT_READ) = 0
62. prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
63. munmap(0x7fd55c719000, 61711) = 0
64. getRandom("/\x9f\x8b\x75\xe2\xd7\x88\x14\x59", 8, GRND_NONBLOCK) = 8
65. brk(NULL) = 0x5566b6305000
66. brk(0x5566b6326000) = 0x5566b6326000
67. futex(0x7fd55c62977c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
68. openat(AT_FDCWD, "/dev/shm/qq", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0600) = 3
69. ftruncate(3, 1024) = 0
70. mmap(NULL, 1024, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fd55c762000
71. newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH)
    = 0
72. read(0, q qqqq
73. "q qqqq \n", 1024) = 8
74. clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
    child_tidptr=0x7fd55c6f8690) = 12246
75. wait4(-1, NULL, 0, NULL) = 12246
76. --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=12246, si_uid=1000, si_status=0,
    si_etime=0, si_stime=0} ---

```

```
77. newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH)
    = 0
78. write(1, "Q_QQQQ_\n", 8Q_QQQQ_
79. )      = 8
80. munmap(0x7fd55c762000, 1024)      = 0
81. unlink("/dev/shm/qq")            = 0
82. exit_group(0)                    = ?
83. +++ exited with 0 +++
```

Вывод

В процессе выполнения данной лабораторной работы я познакомился с утилитой отслеживания системных вызовов `strace Linux`.

Используя ее, можно понять, к каким файлам обращается программа, какие сетевые порты она использует, какие ресурсы ей нужны, а также какие ошибки возвращает ей система. Это помогает разобраться в особенностях работы программы и лучше понять причину ошибки.

Несмотря на то, что эта утилита редко используется при дебаге лабораторных работ, я уверен, что опыт работы с ней пригодится мне в дальнейшей работе с более объемными проектами.