

Лабораторная работа №8

по курсу "Операционные системы"

Выполнил: Юнусов Рустам М8О-310Б-22. 5 Вариант

Преподаватель: Миронов Евгений Сергеевич

Задание:

При выполнении лабораторных работ по курсу ОС необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР.

По итогам выполнения всех лабораторных работ отчет по данной ЛР должен содержать краткую сводку по исследованию написанных программ.

Пример работы

Лабораторная работа 3 – взаимодействие между потоками

```
~/codes_rd/lab4_operation_systems_linux$ strace ./main
execve("./main", [ "./main" ], 0x7ffe185fbe90 /* 25 vars */) = 0
brk(NULL)                               = 0x61e877dc0000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x719bccccc000
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=59211, ...}) = 0
mmap(NULL, 59211, PROT_READ, MAP_PRIVATE, 3, 0) = 0x719bcccbd000
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\3\0>\0\1\0\0\0\220\243\2\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"..., 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"..., 784,
64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x719bccca00000
mmap(0x719bccca28000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x719bccca28000
```

```

mmap(0x719bccbb0000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x719bccbb0000
mmap(0x719bccbfff000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x719bccbfff000
mmap(0x719bccc05000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x719bccc05000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x719bcccba000
arch_prctl(ARCH_SET_FS, 0x719bcccba740) = 0
set_tid_address(0x719bcccbaa10) = 1091678
set_robust_list(0x719bcccbaa20, 24) = 0
rseq(0x719bcccb060, 0x20, 0, 0x53053053) = 0
mprotect(0x719bccbfff000, 16384, PROT_READ) = 0
mprotect(0x61e8774a0000, 4096, PROT_READ) = 0
mprotect(0x719bccd04000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x719bcccbd000, 59211) = 0
openat(AT_FDCWD, "shared_mem_file", O_RDWR|O_CREAT, 0666) = 3
ftruncate(3, 4) = 0
mmap(NULL, 4, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x719bccccb000
getrandom("\x27\x1c\xb1\x87\xee\xfb\x2e\xc5", 8, GRND_NONBLOCK) = 8
newfstatat(AT_FDCWD, "/dev/shm/sem.9B65Il", 0x7ffc981f3ed0,
AT_SYMLINK_NOFOLLOW) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/dev/shm/sem.9B65Il",
O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666) = 4
write(4,
"\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\0\0",
32) = 32
mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x719bcccca000
link("/dev/shm/sem.9B65Il", "/dev/shm/sem.mysemaphore") = -1 EEXIST
(File exists)
munmap(0x719bcccca000, 32) = 0
unlink("/dev/shm/sem.9B65Il") = 0
close(4) = 0
unlink("/dev/shm/sem.mysemaphore") = 0
openat(AT_FDCWD, "/dev/shm/sem.mysemaphore",
O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT (No such file or directory)
getrandom("\x4c\xc7\xe5\x65\xca\xd8\xc9\x31", 8, GRND_NONBLOCK) = 8
newfstatat(AT_FDCWD, "/dev/shm/sem.ykQAnp", 0x7ffc981f3ed0,
AT_SYMLINK_NOFOLLOW) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/dev/shm/sem.ykQAnp",
O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666) = 4
write(4,
"\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\0\0",
32) = 32
mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x719bcccca000
link("/dev/shm/sem.ykQAnp", "/dev/shm/sem.mysemaphore") = 0
fstat(4, {st_mode=S_IFREG|0664, st_size=32, ...}) = 0
getrandom("\x6e\xd3\xa3\xed\xeb\x47\x55\x29", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x61e877dc0000
brk(0x61e877de1000) = 0x61e877de1000

```

```

unlink("/dev/shm/sem.ykQAnp")      = 0
close(4)                            = 0
clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x719bcccb000) = 1091679
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\207\320\270\321\201\320\273\320\276:\n", 27Введите число:
) = 27
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
read(0, 12
"12\n", 1024)                        = 3
futex(0x719bcccb000, FUTEX_WAKE, 1)   = 1
wait4(1091679, 0x7ffc981f4258, WNOHANG, NULL) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\207\320\270\321\201\320\273\320\276:\n", 27Введите число:
) = 27
read(0, 14
"14\n", 1024)                        = 3
futex(0x719bcccb000, FUTEX_WAKE, 1)   = 1
wait4(1091679, 0x7ffc981f4258, WNOHANG, NULL) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\207\320\270\321\201\320\273\320\276:\n", 27Введите число:
) = 27
read(0, 17
"17\n", 1024)                        = 3
futex(0x719bcccb000, FUTEX_WAKE, 1)   = 1
wait4(1091679, 0x7ffc981f4258, WNOHANG, NULL) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\207\320\270\321\201\320\273\320\276:\n", 27Введите число:
) = 27
Дочерний процесс завершен.
read(0, 0x61e877dc0710, 1024)        = ? ERESTARTSYS (To be restarted
if SA_RESTART is set)
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=1091679,
si_uid=143129039, si_status=0, si_utime=0, si_stime=0} ---
read(0,
"\n", 1024)                          = 1
wait4(1091679, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], WNOHANG, NULL) =
1091679
munmap(0x719bcccb000, 32)             = 0
unlink("/dev/shm/sem.mysemaphore")    = 0
munmap(0x719bcccb000, 4)              = 0
close(3)                              = 0
unlink("shared_mem_file")             = 0
write(1,
"\320\240\320\276\320\264\320\270\321\202\320\265\320\273\321\214\321\20
1\320\272\320\270\320\271 \320\277\321\200\320\276\321"...,
58Родительский процесс завершен.
) = 58
exit_group(0)                         = ?
+++ exited with 0 +++

```

Вывод

Во время этой лабораторной работы я познакомился с утилитой `strace` в Linux (почему-то нет на macOS: ()), которая отслеживает системные вызовы.

С её помощью можно увидеть, к каким файлам обращается программа, какие сетевые порты она использует, какие ресурсы ей нужны и какие ошибки она получает от системы. Это помогает понять, как работает программа и найти причину возможных ошибок.

Хотя `strace` используют все реже при отладке, я думаю, что опыт работы с ней будет полезен мне в дальнейшем при работе с большими проектами.