

**Московский авиационный институт
(национальный исследовательский университет)**

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»
Дисциплина «Операционные системы»

Лабораторная работа №1

Тема: Диагностика работы программного обеспечения

Студент: Инютин М. А.
Группа: М8О-207Б-19
Преподаватель: Миронов Е. С.
Дата:
Оценка:

1. Постановка задачи

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 отчет по данной должен содержать краткую сводку по исследованию последующих ЛР.

Используемая утилита для отслеживания системных вызовов: strace.

2. Лабораторная работы №2

В работе я использовал системные вызовы *fopen*, *pipe*, *fork*, *dup2* и *execl*. *strace* показал все вызовы чтения и записи (вызовы *read* и *write* возвращают число прочитанных/записанных байт), как и во всех следующих лабораторных. Так же видно, что используется *pipe*, а *fork* вызывает *clone* для создания дочернего процесса.

```
read(0, "input.txt\n", 4096)           = 10
lseek(0, -1, SEEK_CUR)                 = 9
openat(AT_FDCWD, "input.txt", O_RDONLY) = 3
dup3(3, 0, 0)                          = 0
close(3)                               = 0
pipe([3, 4])                           = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|
CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7fa983d47810) = 21082
close(4)                               = 0
read(3, "\322\0\0\0", 4)               = 4
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...})
= 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=21082,
si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---
write(1, "210\n", 4210)
)                                       = 4
read(3, "\0\0\0\0", 4)                 = 4
write(1, "0\n", 20)
)                                       = 2
read(3, "\1\0\0\0", 4)                 = 4
write(1, "1\n", 21)
)                                       = 2
read(3, "\377\377\377\177", 4)         = 4
write(1, "2147483647\n", 112147483647)
)                                       = 11
read(3, "\0\0\0\200", 4)               = 4
write(1, "-2147483648\n", 12-2147483648)
)                                       = 12
read(3, "7\0\0\0", 4)                  = 4
write(1, "55\n", 355)
)                                       = 3
read(3, "", 4)                         = 0
close(3)                              = 0
exit_group(0)                          = ?
+++ exited with 0 +++
```

3. Лабораторная работа №3

Я использовал следующие системные три функции: *pthread_create* и *pthread_join* для работы с потоками, *clock* для измерения времени работы программы. Так как создание потока происходит с помощью *clone* с особыми флагами, то *strace* показал создание двух потоков как вызов *clone* два раза.

```
write(1, "Thread number: 2\n", 17Thread number: 2
) = 17
fstat(0, {st_mode=S_IFREG|0664, st_size=27, ...}) = 0
read(0, "1 5\n1 0\n5 0\n4 1\n3 2\n2 3\n1 4", 4096) = 27
read(0, "", 4096) = 0
clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0,
tv_nsec=1044475}) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|
MAP_STACK, -1, 0) = 0x7fe3233ea000
mprotect(0x7fe3233eb000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7fe323be9fb0, flags=CLONE_VM|CLONE_FS|
CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|
CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[30051],
tls=0x7fe323bea700, child_tidptr=0x7fe323bea9d0) = 30051
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|
MAP_STACK, -1, 0) = 0x7fe322be9000
mprotect(0x7fe322bea000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7fe3233e8fb0, flags=CLONE_VM|CLONE_FS|
CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|
CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[30052],
tls=0x7fe3233e9700, child_tidptr=0x7fe3233e99d0) = 30052
clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0,
tv_nsec=1158424}) = 0
write(1, "Execution time 0.114000 ms\n", 27Execution time 0.114000
ms
) = 27
write(1, "5 0\n", 45 0
) = 4
write(1, "4 1\n", 44 1
) = 4
write(1, "3 2\n", 43 2
) = 4
write(1, "2 3\n", 42 3
) = 4
write(1, "1 4\n", 41 4
) = 4
exit_group(0) = ?
+++ exited with 0 +++
```

4. Лабораторная работа №4

Для открытия файла используется вызов `openat`. Так как общие файлы расположены в `/dev/shm`, то системные вызовы ссылаются на этот путь. `ftruncate` для общего файла использует одноимённый вызов `ftruncate`. Отображение файла в память использует системный вызов `mmap` и `mmap`. Снова для создания процесса система вызывает `clone`. Работа с общим мьютексом происходит через вызов `futex`, который возвращает `EAGAIN` в случае, когда нужно блокировать процесс. Удаление общих файлов осуществляется `shm_unlink`.

```
read(0, "input.txt\n", 4096) = 10
openat(AT_FDCWD, "input.txt", O_RDONLY) = 3
statfs("/dev/shm/", {f_type=TMPFS_MAGIC, f_bsize=4096,
f_blocks=4090665, f_bfree=3952829, f_bavail=3952829,
f_files=4090665, f_ffree=4089358, f_fsid={val=[0, 0]},
f_namelen=255, f_frsize=4096, f_flags=ST_VALID|ST_NOSUID|
ST_NODEV}) = 0
futex(0x7fdeab685390, FUTEX_WAKE_PRIVATE, 2147483647) = 0
openat(AT_FDCWD, "/dev/shm/shared_file", O_RDWR|O_CREAT|
O_NOFOLLOW|O_CLOEXEC, 0700) = 4
ftruncate(4, 16) = 0
openat(AT_FDCWD, "/dev/shm/shared_mutex", O_RDWR|O_CREAT|
O_NOFOLLOW|O_CLOEXEC, 0700) = 5
ftruncate(5, 40) = 0
mmap(NULL, 40, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0) =
0x7fdeab6dc000
openat(AT_FDCWD, "/dev/shm/shared_cond", O_RDWR|O_CREAT|
O_NOFOLLOW|O_CLOEXEC, 0700) = 6
ftruncate(6, 48) = 0
mmap(NULL, 48, PROT_READ|PROT_WRITE, MAP_SHARED, 6, 0) =
0x7fdeab6af000
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|
CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7fdeab46ea10) = 21600
mmap(NULL, 16, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) =
0x7fdeab6ae000
futex(0x7fdeab6af028, FUTEX_WAIT, 0, NULL) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) =
0
write(1, "210\n", 4210
) = 4
futex(0x7fdeab6af02c, FUTEX_WAKE, 1) = 1
futex(0x7fdeab6dc000, FUTEX_WAKE, 1) = 0
futex(0x7fdeab6af028, FUTEX_WAIT, 0, NULL) = -1 EAGAIN (Resource
temporarily unavailable)
write(1, "0\n", 20
) = 2
futex(0x7fdeab6af02c, FUTEX_WAKE, 1) = 1
futex(0x7fdeab6dc000, FUTEX_WAKE, 1) = 0
write(1, "1\n", 21
) = 2
```

```

futex(0x7fdeab6af028, FUTEX_WAKE, 1)      = 1
futex(0x7fdeab6af02c, FUTEX_WAIT, 0, NULL) = -1 EAGAIN (Resource
temporarily unavailable)
write(1, "2147483647\n", 112147483647
)
      = 11
futex(0x7fdeab6af028, FUTEX_WAKE, 1)      = 1
futex(0x7fdeab6dc000, FUTEX_WAKE, 1)      = 0
write(1, "-2147483648\n", 12-2147483648
)
      = 12
futex(0x7fdeab6af02c, FUTEX_WAKE, 1)      = 1
futex(0x7fdeab6af028, FUTEX_WAIT, 0, NULL) = -1 EAGAIN (Resource
temporarily unavailable)
write(1, "55\n", 355
)
      = 3
futex(0x7fdeab6af02c, FUTEX_WAKE, 1)      = 1
futex(0x7fdeab6dc000, FUTEX_WAKE, 1)      = 0
munmap(0x7fdeab6ae000, 16)                = 0
munmap(0x7fdeab6dc000, 40)                = 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=21600,
si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---
munmap(0x7fdeab6af000, 48)                = 0
unlink("/dev/shm/shared_file")            = 0
unlink("/dev/shm/shared_mutex")          = 0
unlink("/dev/shm/shared_cond")           = 0
close(3)                                  = 0
lseek(0, -1, SEEK_CUR)                    = 9
exit_group(0)                             = ?
+++ exited with 0 +++

```

5. Лабораторная работа №5

Для работы с динамическими библиотеками я использовал *dlopen* и *dlsym*. *strace* показал, что сначала программа открывает библиотеку по указанному пути с использованием *getcwd*, а потом отображает части файла в память вызовами *openat* и *mmap*.

```
openat(AT_FDCWD, "./libimpl1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\20\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=18088, ...}) = 0
getcwd("/home/engineerxl/Study/OS/lab5", 128) = 31
mmap(NULL, 16424, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fb4188de000
mmap(0x7fb4188df000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fb4188df000
mmap(0x7fb4188e0000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e0000
mmap(0x7fb4188e1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e1000
close(3) = 0
mprotect(0x7fb4188e1000, 4096, PROT_READ) = 0
fstat(0, {st_mode=S_IFREG|0664, st_size=34, ...}) = 0
read(0, "1\n8 4\n0\n1\n8 4\n0\n2\n2\n3 1\n0\n2\n2\n3 "... , 4096) =
34
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) =
0
write(1, "GCD(8, 4) = 4\n", 14GCD(8, 4) = 4
) = 14
munmap(0x7fb4188de000, 16424) = 0
openat(AT_FDCWD, "./libimpl2.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\20\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=19736, ...}) = 0
getcwd("/home/engineerxl/Study/OS/lab5", 128) = 31
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fb4188de000
mmap(0x7fb4188df000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fb4188df000
mmap(0x7fb4188e0000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e0000
mmap(0x7fb4188e1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e1000
close(3) = 0
mprotect(0x7fb4188e1000, 4096, PROT_READ) = 0
write(1, "GCD(8, 4) = 4\n", 14GCD(8, 4) = 4
) = 14
munmap(0x7fb4188de000, 16432) = 0
openat(AT_FDCWD, "./libimpl1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\20\0\0\0\0\0\0"... , 832) = 832
```

```

fstat(3, {st_mode=S_IFREG|0775, st_size=18088, ...}) = 0
getcwd("/home/engineerxl/Study/OS/lab5", 128) = 31
mmap(NULL, 16424, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fb4188de000
mmap(0x7fb4188df000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fb4188df000
mmap(0x7fb4188e0000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e0000
mmap(0x7fb4188e1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e1000
close(3) = 0
mprotect(0x7fb4188e1000, 4096, PROT_READ) = 0
write(1, "Sorted: 1 3 \n", 13Sorted: 1 3
) = 13
munmap(0x7fb4188de000, 16424) = 0
openat(AT_FDCWD, "./libimpl2.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\`
20\0\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0775, st_size=19736, ...}) = 0
getcwd("/home/engineerxl/Study/OS/lab5", 128) = 31
mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fb4188de000
mmap(0x7fb4188df000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7fb4188df000
mmap(0x7fb4188e0000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e0000
mmap(0x7fb4188e1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7fb4188e1000
close(3) = 0
mprotect(0x7fb4188e1000, 4096, PROT_READ) = 0
write(1, "Sorted: 1 3 \n", 13Sorted: 1 3
) = 13
read(0, "", 4096) = 0
munmap(0x7fb4188de000, 16432) = 0
exit_group(0) = ?
+++ exited with 0 +++

```


6. Лабораторная работа №6

Программа использует сокеты из библиотеки *ZeroMQ* с протоколом *TCP* и устанавливает значения *ZMQ_RECVTIMEO* и *ZMQ_SENDTIMEO* на 1000 мс, поэтому *strace* показывает *poll* с временем ожидания 1000 для функций *zmq_msg_recv* и *zmq_msg_send*. *zmq_bind* использует системные вызовы *bind* для привязка и *listen* для просмотра соединений.

```
socket(AF_INET, SOCK_STREAM|SOCK_CLOEXEC, IPPROTO_TCP) = 9
setsockopt(9, SOL_SOCKET, SO_REUSEADDR, [1], 4) = 0
bind(9, {sa_family=AF_INET, sin_port=htons(8001),
sin_addr=inet_addr("0.0.0.0")}, 16) = 0
listen(9, 100) = 0
getsockname(9, {sa_family=AF_INET, sin_port=htons(8001),
sin_addr=inet_addr("0.0.0.0")}, [128->16]) = 0
getsockname(9, {sa_family=AF_INET, sin_port=htons(8001),
sin_addr=inet_addr("0.0.0.0")}, [128->16]) = 0
write(6, "\1\0\0\0\0\0\0\0", 8) = 8
write(8, "\1\0\0\0\0\0\0\0", 8) = 8
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|
CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7f26213988d0) = 54099
poll([{fd=8, events=POLLIN}], 1, 0) = 1 ([{fd=8,
revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8) = 8
poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)
poll([{fd=8, events=POLLIN}], 1, 1000) = 1 (OK: 54099)
) = 1 ([{fd=8, revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8) = 8
poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)
poll([{fd=8, events=POLLIN}], 1, 1000) = 1 ([{fd=8,
revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8) = 8
poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)
poll([{fd=8, events=POLLIN}], 1, 1000) = 1 ([{fd=8,
revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8) = 8
poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) =
0
write(1, "OK: 1\n", 6) = 6
write(6, "\1\0\0\0\0\0\0\0", 8) = 8
poll([{fd=8, events=POLLIN}], 1, 1000) = 1 ([{fd=8,
revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8) = 8
poll([{fd=8, events=POLLIN}], 1, 0) = 0 (Timeout)
write(6, "\1\0\0\0\0\0\0\0", 8) = 8
poll([{fd=8, events=POLLIN}], 1, 1000) = 1 ([{fd=8,
revents=POLLIN}])
```

```

read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)       = 0 (Timeout)
write(6, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 1000)    = 1 ([{fd=8,
revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)       = 0 (Timeout)
write(6, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 1000)    = 1 ([{fd=8,
revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)       = 0 (Timeout)
write(1, "OK\n", 30K
)          = 3
read(0, "", 4096)                          = 0
write(6, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 1000)    = 1 ([{fd=8,
revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)       = 0 (Timeout)
write(4, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=3, events=POLLIN}], 1, -1)      = 1 ([{fd=3,
revents=POLLIN}])
read(3, "\1\0\0\0\0\0\0\0", 8)          = 8
write(6, "\1\0\0\0\0\0\0\0", 8)          = 8
futex(0x7f2620b7f9d0, FUTEX_WAIT, 54098, NULL) = -1 EAGAIN
(Resource temporarily unavailable)
close(7)                                    = 0
close(6)                                    = 0
close(5)                                    = 0
close(4)                                    = 0
close(3)                                    = 0
write(1, "OK\n", 30K
)          = 3
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=54099,
si_uid=1000, si_status=0, si_etime=0, si_stime=0} ---
exit_group(0)                              = ?
+++ exited with 0 +++

```

7. Выводы

Во время выполнения работы я узнал, что утилита `strace` позволяет отслеживать обращения программы к операционной системе и таким образом искать возможные ошибки, что было актуально при отладке лабораторных работ 6-8.