Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу

«Операционные системы»

Студент: Знай Артемий Олегович
Группа: М8О–201Б–21
Вариант: 0
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

Репозиторий

https://github.com/znako/OS_LABS

Постановка задачи

Цель работы

Приобретение практических навыков диагностики работы программного обеспечения на примере 4 лабораторной работы.

Задание

Провести диагностику работы 4 лабораторной работы при помощи strace, объяснить результат работы strace.

Вариант 18: Нечетные строки отправляются в child1 четные в child2, дочерние процессы удаляют все гласные из строк.

Вывод strace

```
znako@znako-VirtualBox:~/UtoW/OS_LABS/build/lab4$ strace -f -o write.log ./lab4
o1.txt
o2.txt
jaghsa
kasgkjsjg
aksg
znako@znako-VirtualBox:~/UtoW/OS_LABS/build/lab4$ cat write.log
2173  execve("./lab4", ["./lab4"], 0x7ffde0a34a48 /* 58 vars */) = 0
2173  brk(NULL)                         = 0x55f681e41000
2173  arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd1b50ab70) = -1 EINVAL (Invalid argument)
2173  access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
2173  openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
2173  fstat(3, {st_mode=S_IFREG|0644, st_size=74553, ...}) = 0
2173  mmap(NULL, 74553, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f7adcdb9000
2173  close(3)                          = 0
2173  openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
2173  read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220q\0\0\0\0\0\0"..., 832) = 832
2173  pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\360\2300%\360\340\363'\246\332u/\364\377\246u"..., 68, 824) = 68
2173  fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0
2173  mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f7adcdb7000
2173  pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\360\2300%\360\340\363'\246\332u/\364\377\246u"..., 68, 824) = 68
2173  mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f7adcd94000
```

2173 mmap(0x7f7adcd9a000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7f7adcd9a000
2173 mmap(0x7f7adcdab000, 24576, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17000) = 0x7f7adcdab000
2173 mmap(0x7f7adcdb1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000) = 0x7f7adcdb1000
2173 mmap(0x7f7adcdb3000, 13432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f7adcdb3000
2173 close(3) = 0
2173 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
2173 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360A\2\0\0\0\0"..., 832) = 832
2173 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
2173 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
2173 pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\237\333t\347\262\27\320l\223\27*\202C\370T\177"..., 68, 880) = 68
2173 fstat(3, {st_mode=S_IFREG|0755, st_size=2029560, ...}) = 0
2173 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
2173 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
2173 pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\237\333t\347\262\27\320l\223\27*\202C\370T\177"..., 68, 880) = 68
2173 mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f7adcba2000
2173 mmap(0x7f7adcbc4000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7f7adcbc4000
2173 mmap(0x7f7adcd3c000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19a000) = 0x7f7adcd3c000
2173 mmap(0x7f7adcd8a000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f7adcd8a000
2173 mmap(0x7f7adcd90000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f7adcd90000
2173 close(3) = 0
2173 mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f7adcb9f000
2173 arch_prctl(ARCH_SET_FS, 0x7f7adcb9f740) = 0
2173 mprotect(0x7f7adcd8a000, 16384, PROT_READ) = 0
2173 mprotect(0x7f7adcdb1000, 4096, PROT_READ) = 0
2173 mprotect(0x55f67ffc8000, 4096, PROT_READ) = 0
2173 mprotect(0x7f7adcdf9000, 4096, PROT_READ) = 0

```
2173  munmap(0x7f7adcdb9000, 74553)     = 0
2173  set_tid_address(0x7f7adcb9fa10)   = 2173
2173  set_robust_list(0x7f7adcb9fa20, 24) = 0
2173                  rt_sigaction(SIGRTMIN,           {sa_handler=0x7f7adcd9abf0,           sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f7adcda83c0}, NULL, 8) = 0
2173                   rt_sigaction(SIGRT_1,           {sa_handler=0x7f7adcd9ac90,           sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f7adcda83c0}, NULL, 8) = 0
2173  rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
2173  prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) =
0
2173  brk(NULL)                         = 0x55f681e41000
2173  brk(0x55f681e62000)               = 0x55f681e62000
2173  fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
2173  read(0, "o1.txt\n", 1024)         = 7
2173  read(0, "o2.txt\n", 1024)         = 7
2173  unlink("o1.txt")                  = 0
2173  unlink("o2.txt")                  = 0
2173  openat(AT_FDCWD, "o1.txt", O_WRONLY|O_CREAT, 0700) = 3
2173  openat(AT_FDCWD, "o2.txt", O_WRONLY|O_CREAT, 0700) = 4
2173   statfs("/dev/shm/", {f_type=TMPFS_MAGIC, f_bsize=4096, f_blocks=252845, f_bfree=252845,
f_bavail=252845, f_files=252845, f_ffree=252844, f_fsid={val=[2331749600, 4026586175]},
f_namelen=255, f_frsize=4096, f_flags=ST_VALID|ST_NOSUID|ST_NODEV}) = 0
2173  futex(0x7f7adcdb6390, FUTEX_WAKE_PRIVATE, 2147483647) = 0
2173  openat(AT_FDCWD, "/dev/shm/sem.semaphore1", O_RDWR|O_NOFOLLOW) = -1 ENOENT (No
such file or directory)
2173  getpid()                          = 2173
2173  lstat("/dev/shm/HTVC6v", 0x7ffd1b50a830) = -1 ENOENT (No such file or directory)
2173  openat(AT_FDCWD, "/dev/shm/HTVC6v", O_RDWR|O_CREAT|O_EXCL, 0600) = 5
2173  write(5, "\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\0\0", 32) = 32
2173  mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0) = 0x7f7adcdf8000
2173  link("/dev/shm/HTVC6v", "/dev/shm/sem.semaphore1") = 0
2173  fstat(5, {st_mode=S_IFREG|0600, st_size=32, ...}) = 0
2173  unlink("/dev/shm/HTVC6v")         = 0
2173  close(5)                          = 0
2173  openat(AT_FDCWD, "/dev/shm/sem.semaphore2", O_RDWR|O_NOFOLLOW) = -1 ENOENT (No
such file or directory)
2173  getpid()                          = 2173
2173  lstat("/dev/shm/r0fS2u", 0x7ffd1b50a830) = -1 ENOENT (No such file or directory)
```

```
2173  openat(AT_FDCWD, "/dev/shm/r0fS2u", O_RDWR|O_CREAT|O_EXCL, 0600) = 5
2173  write(5, "\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\0\0", 32) = 32
2173  mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0) = 0x7f7adcdcb000
2173  link("/dev/shm/r0fS2u", "/dev/shm/sem.semaphore2") = 0
2173  fstat(5, {st_mode=S_IFREG|0600, st_size=32, ...}) = 0
2173  unlink("/dev/shm/r0fS2u")         = 0
2173  close(5)                          = 0
2173  mmap(NULL, 512, PROT_READ|PROT_WRITE, MAP_SHARED|MAP_ANONYMOUS, -1, 0) =
0x7f7adcdca000
2173  mmap(NULL, 512, PROT_READ|PROT_WRITE, MAP_SHARED|MAP_ANONYMOUS, -1, 0) =
0x7f7adcdc9000
2173                                                    clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f7adcb9fa10) = 2174
2173                                                    clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>
2174  set_robust_list(0x7f7adcb9fa20, 24) = 0
2173  <... clone resumed>, child_tidptr=0x7f7adcb9fa10) = 2175
2175  set_robust_list(0x7f7adcb9fa20, 24 <unfinished ...>
2173  read(0,  <unfinished ...>
2174  dup2(3, 1 <unfinished ...>
2175  <... set_robust_list resumed>)    = 0
2174  <... dup2 resumed>)               = 1
2175  dup2(4, 1 <unfinished ...>
2174  openat(AT_FDCWD, "/dev/shm/sem.semaphore1", O_RDWR|O_NOFOLLOW <unfinished ...>
2175  <... dup2 resumed>)               = 1
2174  <... openat resumed>)             = 5
2175  openat(AT_FDCWD, "/dev/shm/sem.semaphore2", O_RDWR|O_NOFOLLOW <unfinished ...>
2174  fstat(5,  <unfinished ...>
2175  <... openat resumed>)             = 5
2174  <... fstat resumed>{st_mode=S_IFREG|0600, st_size=32, ...}) = 0
2175  fstat(5,  <unfinished ...>
2174  close(5 <unfinished ...>
2175  <... fstat resumed>{st_mode=S_IFREG|0600, st_size=32, ...}) = 0
2174  <... close resumed>)              = 0
2175  close(5 <unfinished ...>
2174      futex(0x7f7adcdf8000,  FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,  0,  NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
```

```
2175  <... close resumed>)              = 0
2175      futex(0x7f7adcdcb000,  FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,  0,  NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
2173  <... read resumed>"jaghsa\n", 1024) = 7
2173  futex(0x7f7adcdcb000, FUTEX_WAKE, 1) = 1
2175  <... futex resumed>)              = 0
2173  read(0,  <unfinished ...>
2175  write(1, "j", 1)                 = 1
2175  write(1, "g", 1)                 = 1
2175  write(1, "h", 1)                 = 1
2175  write(1, "s", 1)                 = 1
2175  write(1, "\n", 1)                = 1
2175      futex(0x7f7adcdcb000,  FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,  0,  NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
2173  <... read resumed>"kasgkjsjg\n", 1024) = 10
2173  futex(0x7f7adcdf8000, FUTEX_WAKE, 1) = 1
2174  <... futex resumed>)              = 0
2173  read(0,  <unfinished ...>
2174  write(1, "k", 1)                 = 1
2174  write(1, "s", 1)                 = 1
2174  write(1, "g", 1)                 = 1
2174  write(1, "k", 1)                 = 1
2174  write(1, "j", 1)                 = 1
2174  write(1, "s", 1)                 = 1
2174  write(1, "j", 1)                 = 1
2174  write(1, "g", 1)                 = 1
2174  write(1, "\n", 1)                = 1
2174      futex(0x7f7adcdf8000,  FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,  0,  NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
2173  <... read resumed>"aksg\n", 1024) = 5
2173  futex(0x7f7adcdcb000, FUTEX_WAKE, 1) = 1
2175  <... futex resumed>)              = 0
2173  read(0,  <unfinished ...>
2175  write(1, "k", 1)                 = 1
2175  write(1, "s", 1)                 = 1
2175  write(1, "g", 1)                 = 1
2175  write(1, "\n", 1)                = 1
```

```
2175  futex(0x7f7adcdcb000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
2173  <... read resumed>"", 1024)       = 0
2173  futex(0x7f7adcdf8000, FUTEX_WAKE, 1) = 1
2174  <... futex resumed>)              = 0
2173  futex(0x7f7adcdcb000, FUTEX_WAKE, 1) = 1
2175  <... futex resumed>)              = 0
2174  close(3 <unfinished ...>
2173  munmap(0x7f7adcdca000, 512 <unfinished ...>
2175  close(4 <unfinished ...>
2173  <... munmap resumed>)             = 0
2174  <... close resumed>)              = 0
2173  munmap(0x7f7adcdc9000, 512 <unfinished ...>
2175  <... close resumed>)              = 0
2173  <... munmap resumed>)             = 0
2174  munmap(0x7f7adcdca000, 512 <unfinished ...>
2173  munmap(0x7f7adcdf8000, 32 <unfinished ...>
2175  munmap(0x7f7adcdca000, 512 <unfinished ...>
2173  <... munmap resumed>)             = 0
2175  <... munmap resumed>)             = 0
2174  <... munmap resumed>)             = 0
2173  munmap(0x7f7adcdcb000, 32 <unfinished ...>
2175  munmap(0x7f7adcdc9000, 512 <unfinished ...>
2173  <... munmap resumed>)             = 0
2175  <... munmap resumed>)             = 0
2174  munmap(0x7f7adcdc9000, 512 <unfinished ...>
2173  unlink("/dev/shm/sem.semaphore1" <unfinished ...>
2175  munmap(0x7f7adcdf8000, 32 <unfinished ...>
2173  <... unlink resumed>)             = 0
2175  <... munmap resumed>)             = 0
2174  <... munmap resumed>)             = 0
2173  unlink("/dev/shm/sem.semaphore2" <unfinished ...>
2175  unlink("/dev/shm/sem.semaphore1" <unfinished ...>
2173  <... unlink resumed>)             = 0
2175  <... unlink resumed>)             = -1 ENOENT (No such file or directory)
2174  munmap(0x7f7adcdcb000, 32 <unfinished ...>
2173  exit_group(0 <unfinished ...>
2175  unlink("/dev/shm/sem.semaphore2" <unfinished ...>
```

```
2173  <... exit_group resumed>)         = ?
2175  <... unlink resumed>)             = -1 ENOENT (No such file or directory)
2174  <... munmap resumed>)             = 0
2173  +++ exited with 0 +++
2175  exit_group(0)                     = ?
2174  unlink("/dev/shm/sem.semaphore1" <unfinished ...>
2175  +++ exited with 0 +++
2174  <... unlink resumed>)             = -1 ENOENT (No such file or directory)
2174  unlink("/dev/shm/sem.semaphore2") = -1 ENOENT (No such file or directory)
2174  exit_group(0)                     = ?
2174  +++ exited with 0 +++
```

Описание работы

execve() – исполняет программу. Возвращает 0 – успешное выполнение.

brk(NULL) - Устанавливает конец сегмента данных в значение NULL, возвращает указатель на начало новой области памяти

access() - Проверяет на существование и на наличие прав на чтение, возвращает -1 – или не существует или нет прав на чтение, errno устанавливается в ENOENT (компонент пути не существует или является "висячей" символической ссылкой).

openat() - Открывает относительно дескриптора указанного каталога с правами доступа. Возвращает новый файловый дескриптор .

fstat() - Заполняет структуру указанную вторым аргументом fstat информацией об файле с файловым дескриптором (1-й аргумент). Возвращает 0 – успешное выполнение.

mmap() - Создает отображение файла в память. Возвращает указатель на начало отраженной памяти.
close() - Закрывает файл. Возвращает 0 – успешное выполнение.

read() - Читает данные из файла в буффер указанный вторым аргументом. Возвращает число успешно считанных байт.

mprotect() - Контролирует доступ к области памяти. Возвращает 0 – успешное завершение.

arch_prctl() - Устанавливает специфичное для архитектуры состояние. Возвращает 0 – успешное выполнение.

munmap() - Снимает отражение из заданной области памяти. Возвращает 0 – успешное выполнение.

write() Записывает данные из буффера (второй аргумент) в файл. Возвращает число успешно записанных байт.
read() Читает данные в буффер из файл. Возвращает число успешно считанных байт.

clone() - Создает процесс-потомок с фалагами, задает положение стека для процесса-потомка, задает указатель на id. Возвращает pid процесса-потомка.

Т.к. я в strace указал флаг –f, информация о процессах выводилась подробно, рядом с системным вызовом указывался pid процесс, где происходит системный вызов:

Вывод

При помощи strace можно удобно анализировать работу программы, смотреть на различные системные вызовы их параметры, что каждый вызов возвращает, также можно смотреть системные вызовы по процессам, все это помогает искать неполадки в работе программы и устранять их.