Московский авиационный институт
(Национальный исследовательский университет)
Факультет "Информационные технологии и прикладная математика"
Кафедра "Вычислительная математика и программирование"

Лабораторная работа №8 по курсу “Операционные системы”

Студент: Слетюрин Кирилл Сергеевич

Группа: М8О-208Б-22

Преподаватель: Миронов Евгений Сергеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2023

Содержание

1 Репозиторий

https://github.com/kirill483/OS

2 Цель работы

Приобретение практических навыков диагностики работы программного обеспечения.

3 Задание

Продемонстрировать ключевые системные вызовы, используемые в лабораторной работе и то, что их использование соответствует варианту ЛР на примере лабораторной работы №1.

4 Описание strace

Команда strace является инструментом диагностики в Linux. Она перехватывает и записывает любые системные вызовы, выполняемые командой. Кроме того, также записывает любой сигнал Linux, отправляемый процессу. Затем мы можем использовать эту информацию для отладки или диагностики программы.

В самом простом варианте strace запускает переданную команду с её аргументами и выводит в стандартный поток ошибок все системные вызовы команды.

Возможные флаги:

- -k - выводить стек вызовов для отслеживаемого процесса после каждого системного вызова
- -o - выводить всю информацию о системных вызовах не в стандартный поток ошибок, а в файл
- -c - подсчитывать количество ошибок, вызовов и время выполнения для каждого системного вызова
- -T - выводить длительность выполнения системного вызова
- -y - выводить пути для файловых дескрипторов
- -yy - выводить информацию о протоколе для файловых дескрипторов
- -p - указывает pid процесса, к которому следует подключиться
- -f - отслеживать также дочерние процессы, если они будут созданы

5 Демонстрация работы программы

```
kirill@DESKTOP-7E05ERB:/mnt/c/Users/Kirill/OS/lab1/build$ strace -f ./main
execve("./main", ["./main"], 0x7ffed897e198 /* 22 vars */) = 0
brk(NULL)                               = 0x557871f20000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc2b160610) = -1 EINVAL (Invalid argument)
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=66845, ...}) = 0
mmap(NULL, 66845, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fee5b16b000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
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)
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\f\4K\246\21\256\356\256\273\203t\346‘\6
fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fee5
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\f\4K\246\21\256\356\256\273\203t\346‘\6
mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fee5b146000
mmap(0x7fee5b14c000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRI
mmap(0x7fee5b15d000, 24576, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1
mmap(0x7fee5b163000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRI
mmap(0x7fee5b165000, 13432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYM
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\300A\2\0\0\0\0\0"..., 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
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\0", 32,
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\356\276]_K‘\213\212S\354Dkc\230\33\272"
fstat(3, {st_mode=S_IFREG|0755, st_size=2029592, ...}) = 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
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\0", 32,
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\356\276]_K‘\213\212S\354Dkc\230\33\272"
mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fee5af54000
mmap(0x7fee5af76000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYW
mmap(0x7fee5b0ee000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x
mmap(0x7fee5b13c000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWR
mmap(0x7fee5b142000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYM
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fee
arch_prctl(ARCH_SET_FS, 0x7fee5af51740) = 0
mprotect(0x7fee5b13c000, 16384, PROT_READ) = 0
mprotect(0x7fee5b163000, 4096, PROT_READ) = 0
mprotect(0x55787032f000, 4096, PROT_READ) = 0
mprotect(0x7fee5b1a9000, 4096, PROT_READ) = 0
munmap(0x7fee5b16b000, 66845)           = 0
set_tid_address(0x7fee5af51a10)         = 1192
set_robust_list(0x7fee5af51a20, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7fee5b14cbf0, sa_mask=[], sa_flags=SA_RESTORE
rt_sigaction(SIGRT_1, {sa_handler=0x7fee5b14cc90, sa_mask=[], sa_flags=SA_RESTORER
```

```
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) =
pipe([3, 4])                            = 0
pipe([5, 6])                            = 0
brk(NULL)                               = 0x557871f20000
brk(0x557871f41000)                     = 0x557871f41000
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x2), ...}) = 0
read(0, file1.0x557871f20320, 1024)          = ? ERESTARTSYS (To be restarted if
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0, txt0x557871f20320, 1024)           = ? ERESTARTSYS (To be restarted if SA_
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0,
"file1.txt\n", 1024)            = 10
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace
, child_tidptr=0x7fee5af51a10) = 1193
[pid  1193] set_robust_list(0x7fee5af51a20, 24 <unfinished ...>
[pid  1192] close(3)                    = 0
[pid  1192] read(5,  <unfinished ...>
[pid  1193] <... set_robust_list resumed>) = 0
[pid  1193] close(4)                    = 0
[pid  1193] close(5)                    = 0
[pid  1193] dup2(3, 0)                  = 0
[pid  1193] dup2(6, 1)                  = 1
[pid  1193] execve("/mnt/c/Users/Kirill/OS/lab1/build/child", ["file1.txt"], 0x7ff
[pid  1193] brk(NULL)                   = 0x55dd8583f000
[pid  1193] arch_prctl(0x3001 /* ARCH_??? */, 0x7ffffadd9db0) = -1 EINVAL (Invalid
[pid  1193] access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or direct
[pid  1193] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 4
[pid  1193] fstat(4, {st_mode=S_IFREG|0644, st_size=66845, ...}) = 0
[pid  1193] mmap(NULL, 66845, PROT_READ, MAP_PRIVATE, 4, 0) = 0x7f1bcbb7a000
[pid  1193] close(4)                    = 0
[pid  1193] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
[pid  1193] read(4, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0
[pid  1193] pread64(4, "\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
[pid  1193] pread64(4, "\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
[pid  1193] pread64(4, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\356\276]_K‘\213\212S\354Dkc
[pid  1193] fstat(4, {st_mode=S_IFREG|0755, st_size=2029592, ...}) = 0
[pid  1193] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
[pid  1193] pread64(4, "\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
[pid  1193] pread64(4, "\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
[pid  1193] pread64(4, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\356\276]_K‘\213\212S\354Dkc
[pid  1193] mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 4, 0) = 0x7f
[pid  1193] mmap(0x7f1bcb9a8000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIX
[pid  1193] mmap(0x7f1bcbb20000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENY
[pid  1193] mmap(0x7f1bcbb6e000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXE
[pid  1193] mmap(0x7f1bcbb74000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXE
[pid  1193] close(4)                    = 0
[pid  1193] arch_prctl(ARCH_SET_FS, 0x7f1bcbb79540) = 0
[pid  1193] mprotect(0x7f1bcbb6e000, 16384, PROT_READ) = 0
```

```
[pid  1193] mprotect(0x55dd84218000, 4096, PROT_READ) = 0
[pid  1193] mprotect(0x7f1bcbbb8000, 4096, PROT_READ) = 0
[pid  1193] munmap(0x7f1bcbb7a000, 66845) = 0
[pid  1193] brk(NULL)                   = 0x55dd8583f000
[pid  1193] brk(0x55dd85860000)         = 0x55dd85860000
[pid  1193] openat(AT_FDCWD, "file1.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4
[pid  1193] write(1, "\1\0\0\0", 4 <unfinished ...>
[pid  1192] <... read resumed>"\1\0\0\0", 4) = 4
[pid  1193] <... write resumed>)        = 4
[pid  1192] read(0,  <unfinished ...>
[pid  1193] read(0, 4
 <unfinished ...>
[pid  1192] <... read resumed>"4\n", 1024) = 2
[pid  1192] write(4, "\4\0\0\0", 4)     = 4
[pid  1193] <... read resumed>"\4\0\0\0", 4) = 4
[pid  1192] read(5,  <unfinished ...>
[pid  1193] write(1, "\1\0\0\0", 4 <unfinished ...>
[pid  1192] <... read resumed>"\1\0\0\0", 4) = 4
[pid  1193] <... write resumed>)        = 4
[pid  1192] read(0,  <unfinished ...>
[pid  1193] fstat(4, {st_mode=S_IFREG|0777, st_size=0, ...}) = 0
[pid  1193] read(0, 6
 <unfinished ...>
[pid  1192] <... read resumed>"6\n", 1024) = 2
[pid  1192] write(4, "\6\0\0\0", 4)     = 4
[pid  1193] <... read resumed>"\6\0\0\0", 4) = 4
[pid  1192] read(5,  <unfinished ...>
[pid  1193] write(1, "\1\0\0\0", 4 <unfinished ...>
[pid  1192] <... read resumed>"\1\0\0\0", 4) = 4
[pid  1193] <... write resumed>)        = 4
[pid  1192] read(0,  <unfinished ...>
[pid  1193] read(0, -1
 <unfinished ...>
[pid  1192] <... read resumed>"-1\n", 1024) = 3
[pid  1192] write(4, "\377\377\377\377", 4) = 4
[pid  1193] <... read resumed>"\377\377\377\377", 4) = 4
[pid  1192] read(5,  <unfinished ...>
[pid  1193] write(1, "\377\377\377\377", 4 <unfinished ...>
[pid  1192] <... read resumed>"\377\377\377\377", 4) = 4
[pid  1193] <... write resumed>)        = 4
[pid  1192] close(4)                    = 0
[pid  1193] write(4, "4\n6\n", 4 <unfinished ...>
[pid  1192] close(5)                    = 0
[pid  1192] close(6)                    = 0
[pid  1192] lseek(0, -1, SEEK_CUR)      = -1 ESPIPE (Illegal seek)
[pid  1192] exit_group(0)               = ?
[pid  1193] <... write resumed>)        = 4
[pid  1193] close(4 <unfinished ...>
[pid  1192] +++ exited with 0 +++
```

```
<... close resumed>)                    = 0
close(0)                                = 0
close(1)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
kirill@DESKTOP-7E05ERB:/mnt/c/Users/Kirill/OS/lab1/build$
```

1. execve("./main ["./main"], 0x7ffed897e198 /* 22 vars */) = 0: Этот вызов execve, который выполняет программу lab1. Значение 0 означает успешное выполнение.

2. brk(NULL) = 0x56197b5bd000: Этот вызов brk используется для расширения размера кучи программы. Здесь он устанавливает верхний предел кучи на адрес 0x56197b5bd000.

3. openat(AT_FDCWD, "/etc/ld.so.cache O_RDONLY|O_CLOEXEC) = 3: Этот вызов открывает файл /etc/ld.so.cache для чтения. Данный файл содержит кэш динамически загружаемых библиотек, которые используются для быстрого поиска библиотек при выполнении программ.

4. fstat(3, st_mode=S_IFREG|0644, st_size=239963, ...) = 0: Этот вызов получает информацию о файле, который открыт дескриптором 3.

5. mmap(NULL, 239963, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f34fc57e000: Выделение памяти с использованием системного вызова mmap. Этот вызов создает отображение виртуальной памяти для чтения (PROT_READ) размером 239963 байт, начиная с адреса 0x7f34fc57e000. Отображение является частным и открыто только для чтения. Файловый дескриптор 3 указывает на файл, откуда происходит отображение.

6. close(3) = 0: Этот вызов закрывает файловый дескриптор 3 (который был использован для ld.so.cache).

7. read(3, "..."..., 832) = 832: Чтение 832 битов из файла /lib/x86_64-linux-gnu/librt.so.1

8. arch_prctl(ARCH_SET_FS, 0x7f34fc00c740) = 0: Задаёт состояние процесса.

9. mprotect(0x7f34fc348000, 16384, PROT_READ) = 0: Этот вызов изменяет права доступа к памяти. Здесь он делает доступной для чтения область памяти, начинающуюся с адреса 0x7f34fc348000 и имеющую размер 16384 байта.

10. munmap(0x7f34fc57e000, 239963) = 0: Снимает отражение файла или устройства в памяти.

11. set_tid_address(0x7f34fc00ca10) = 16879: Этот вызов устанавливает адрес переменной в адресное пространство потока.

12. prlimit64(0, RLIMIT_STACK, NULL, rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY) = = 0: Этот вызов изменяет ограничения ресурсов процесса. Здесь он изменяет текущий размер стека в 8192*1024 байт и максимальный размер стека в бесконечность.

13. futex(0x7f34fc57b390, FUTEX_WAKE_PRIVATE, 2147483647) = 0: Этот вызов реализует операции с futex (Fast Userspace Mutex). Здесь он пробуждает ожидающий поток (FUTEX_WAKE_PRIVATE).

14. read(0, file.txt "file.txt 1024) = 9: Этот вызов производит чтение из стандартного ввода в буфер размером 1024 байта. Прочитанная строка имеет длину 9 байт.

15. ftruncate(3, 1024) = 0: Этот вызов устанавливает размер файла, связанного с файловым дескриптором 3, в 1024 байта.

16. getpid() = 16879: Получение идентификатора текущего процесса.

17. write(4, "0...0 32) = 32: Запись 32 байт нулей в файл, связанный с файловым дескриптором 4.

18. clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace, child_tidptr=0x7f34fc00ca10) = 16973: Создание нового процесса с помощью системного вызова clone.

6 Выводы

В результате выполнения данной лабораторной работы я ознакомился с таким средством диагностики как strace, с помощью которой можно отследить системные вызовы, выполняемые программой. Я приобрел практические навыки диагностики работы программного обеспечения.