МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

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»

Студент: Власко М. М.
Группа: М8О-208Б-23
Преподаватель: Живалев Е. А.
Дата: _____
Оценка: _____
Подпись: _____

Москва, 2024

**Тема:** Диагностика программного обеспечения

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

**Задачи:**

1. Провести диагностику одной из программ, написанных в ходе работ 1-7, с помощью утилиты *strace.*
2. Рассмотреть использовавшиеся системные вызовы.

**Ход работы:**

strace — это утилита для диагностики и отладки программ в операционных системах семейства Unix/Linux. Она позволяет отслеживать системные вызовы и сигналы, которые приложение выполняет или получает во время своей работы.

Трассировка использования *strace* для лабораторной работы №3*:*

```
execve("./lab3", ["./lab3"], 0x7ffe61ed86f0 /* 27 vars */) = 0

brk(NULL)                               = 0x56005b885000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fc3af705000

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=21863, ...}) = 0

mmap(NULL, 21863, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fc3af6ff000

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\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) = 0x7fc3af4ed000

mmap(0x7fc3af515000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENY-
WRITE, 3, 0x28000) = 0x7fc3af515000

mmap(0x7fc3af69d000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7fc3af69d000

mmap(0x7fc3af6ec000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1fe000) = 0x7fc3af6ec000

mmap(0x7fc3af6f2000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7fc3af6f2000

close(3)                                = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fc3af4ea000

arch_prctl(ARCH_SET_FS, 0x7fc3af4ea740) = 0

set_tid_address(0x7fc3af4eaa10)         = 103631

set_robust_list(0x7fc3af4eaa20, 24)     = 0

rseq(0x7fc3af4eb060, 0x20, 0, 0x53053053) = 0

mprotect(0x7fc3af6ec000, 16384, PROT_READ) = 0

mprotect(0x56005b67b000, 4096, PROT_READ) = 0

mprotect(0x7fc3af73d000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x7fc3af6ff000, 21863)           = 0

unlink("/dev/shm/sem.sem_read1")        = -1 ENOENT (No such file or directory)

unlink("/dev/shm/sem.sem_read2")        = -1 ENOENT (No such file or directory)

unlink("/dev/shm/sem.sem_write")        = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/dev/shm/sem.sem_read1", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

getrandom("\xa6\xb8\xa1\xa0\x65\x86\x66\x87", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.gS68VK", 0x7ffecdbd2130, AT_SYMLINK_NOFOLLOW) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "/dev/shm/sem.gS68VK", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC,
0666) = 3

write(3, "\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

mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fc3af704000

link("/dev/shm/sem.gS68VK", "/dev/shm/sem.sem_read1") = 0
```

```
fstat(3, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0

getrandom("\xc6\xf1\xe0\x5e\x75\x87\xeb\xb7", 8, GRND_NONBLOCK) = 8

brk(NULL)                               = 0x56005b885000

brk(0x56005b8a6000)                     = 0x56005b8a6000

unlink("/dev/shm/sem.gS68VK")           = 0

close(3)                                = 0

openat(AT_FDCWD, "/dev/shm/sem.sem_read2", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

getrandom("\x42\x7e\x20\x5f\xad\x10\x50\xb2", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.wyvHhz", 0x7ffecdbd2130, AT_SYMLINK_NOFOLLOW) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "/dev/shm/sem.wyvHhz", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC,
0666) = 3

write(3, "\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

mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fc3af703000

link("/dev/shm/sem.wyvHhz", "/dev/shm/sem.sem_read2") = 0

fstat(3, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0

unlink("/dev/shm/sem.wyvHhz")           = 0

close(3)                                = 0

openat(AT_FDCWD, "/dev/shm/sem.sem_write", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

getrandom("\xe8\xb3\x43\x59\xe8\x3e\x04\x22", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.IL0W8W", 0x7ffecdbd2130, AT_SYMLINK_NOFOLLOW) = -1
ENOENT (No such file or directory)

openat(AT_FDCWD, "/dev/shm/sem.IL0W8W", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC,
0666) = 3

write(3, "\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

mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fc3af702000

link("/dev/shm/sem.IL0W8W", "/dev/shm/sem.sem_write") = 0

fstat(3, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0

unlink("/dev/shm/sem.IL0W8W")           = 0

close(3)                                = 0

openat(AT_FDCWD, "/dev/shm/memory", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0666) = 3
```

```
ftruncate(3, 1024)                      = 0

mmap(NULL, 1024, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fc3af701000

fstat(0, {st_mode=S_IFCHR|0600, st_rdev=makedev(0x88, 0x4), ...}) = 0

read(0, first

"first\n", 1024)                = 6

read(0, second

"second\n", 1024)               = 7

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fc3af4eaa10) = 103671

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fc3af4eaa10) = 103672

read(0, abc

"abc\n", 1024)                  = 4

futex(0x7fc3af704000, FUTEX_WAKE, 1)    = 1

futex(0x7fc3af702000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY) = 0

read(0, abcd

"abcd\n", 1024)                 = 5

futex(0x7fc3af703000, FUTEX_WAKE, 1)    = 1

futex(0x7fc3af702000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
FUTEX_BITSET_MATCH_ANY) = 0

read(0, q

"q\n", 1024)                    = 2

futex(0x7fc3af704000, FUTEX_WAKE, 1)    = 1

futex(0x7fc3af703000, FUTEX_WAKE, 1)    = 1

wait4(-1, NULL, 0, NULL)                = 103671

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=103672, si_uid=0, si_sta-
tus=0, si_utime=0, si_stime=1 /* 0.01 s */} ---

wait4(-1, NULL, 0, NULL)                = 103672

munmap(0x7fc3af704000, 32)              = 0

unlink("/dev/shm/sem.sem_read1")        = 0

munmap(0x7fc3af703000, 32)              = 0

unlink("/dev/shm/sem.sem_read2")        = 0
```

```
munmap(0x7fc3af702000, 32)              = 0
unlink("/dev/shm/sem.sem_write")        = 0
munmap(0x7fc3af701000, 1024)            = 0
close(3)                                = 0
unlink("/dev/shm/memory")               = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Использовавшиеся системные вызовы:

1. *execve* – Выполняет запуск нового процесса, заменяя текущий процесс указанной программой.
2. *brk* – Управляет границей сегмента данных для выделения памяти в куче.
3. *mmap* – Отображает файлы или устройства в память, а также выделяет анонимную память.
4. *access* – Проверяет доступность файла по заданным правам (чтение, запись, выполнение).
5. *openat* – Открывает файл или каталог, относительно заданного файлового дескриптора.
6. *fstat* – Получает информацию о файле по его дескриптору.
7. *read* – Считывает данные из файла или ввода.
8. *pread64* – Читает данные из файла с определённого смещения, не изменяя позицию указателя.
9. *close* – Закрывает файловый дескриптор.
10. *arch_prctl* – Настраивает параметры архитектуры, например, указывает адрес указателя сегмента FS для текущего потока.
11. *set_tid_address* – Устанавливает адрес для хранения идентификатора потока.
12. *set_robust_list* – Устанавливает указатель на список «робастных мьютексов» (защищённых от прерываний).
13. *rseq* – Регистры последовательных операций для оптимизации потоков.
14. *mprotect* – Изменяет права доступа к памяти (например, разрешение на чтение/запись/выполнение).
15. *prlimit64* – Получает или изменяет текущие ограничения ресурсов (например, размер стека).
16. *munmap* – Освобождает ранее отображённую память.

17. *getrandom* – Генерирует случайные числа из ядра.
18. *clone* – Создаёт новый процесс или поток.
19. *write* – Записывает данные в файл или вывод.
20. *clock_nanosleep* – Приостанавливает выполнение потока на указанное время.
21. *restart_syscall* – Повторяет прерванный системный вызов.
22. *wait4* – Ожидает завершения дочернего процесса, возвращая его статус.
23. *exit_group* – Завершает выполнение всех потоков текущего процесса.
24. *newfstatat* — Проверяет существование и свойства файла или объекта.
25. *ftruncate* — Изменяет размер файла.
26. *futex* — Оперирует синхронизацией между потоками.
27. *link* — Создаёт жёсткую ссылку.
28. *unlink* — Удаляет файл или символическую ссылку.
29. *mmap с флагом MAP_SHARED* — Отображает файл с разделяемым доступом.

**Вывод:** В ходе работы была выполнена диагностика одной из программ, созданных ранее, с использованием инструмента *strace*. Исследование системных вызовов продемонстрировало их важность в работе обеспечении взаимодействия программ с ресурсами операционной системы. В ходе анализа выявлены используемые методы взаимодействия с памятью, синхронизацией процессов, вводом-выводом, обменом сообщениями. Полученные данные подтвердили соответствие работы программы теоретическим ожиданиям и их корректность при работе с различными механизмами операционной системы.