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: Ширшов Д. К.

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"], 0x7fffb84e0560 /\* 27 vars \*/) = 0

brk(NULL) = 0x55ffe7b4a000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3c68611000

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=22843, ...}) = 0

mmap(NULL, 22843, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f3c6860b000

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) = 0x7f3c683f9000

mmap(0x7f3c68421000, 1605632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f3c68421000

mmap(0x7f3c685a9000, 323584, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b0000) = 0x7f3c685a9000

mmap(0x7f3c685f8000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1fe000) = 0x7f3c685f8000

mmap(0x7f3c685fe000, 52624, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3c685fe000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3c683f6000

arch\_prctl(ARCH\_SET\_FS, 0x7f3c683f6740) = 0

set\_tid\_address(0x7f3c683f6a10) = 604

set\_robust\_list(0x7f3c683f6a20, 24) = 0

rseq(0x7f3c683f7060, 0x20, 0, 0x53053053) = 0

mprotect(0x7f3c685f8000, 16384, PROT\_READ) = 0

mprotect(0x55ffad2cc000, 4096, PROT\_READ) = 0

mprotect(0x7f3c68649000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7f3c6860b000, 22843) = 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("\x0d\xff\x41\xe1\x42\x3d\xb3\x92", 8, GRND\_NONBLOCK) = 8

newfstatat(AT\_FDCWD, "/dev/shm/sem.ZdMlaj", 0x7ffcac843760, AT\_SYMLINK\_NOFOLLOW) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/dev/shm/sem.ZdMlaj", O\_RDWR|O\_CREAT|O\_EXCL|O\_NOFOLLOW|O\_CLOEXEC, 0666) = 3

write(3, "\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\0", 32) = 32

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f3c68610000

link("/dev/shm/sem.ZdMlaj", "/dev/shm/sem.sem\_read1") = 0

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=32, ...}) = 0

getrandom("\x33\xd2\x20\x7c\xe8\x48\xa1\xc0", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55ffe7b4a000

brk(0x55ffe7b6b000) = 0x55ffe7b6b000

unlink("/dev/shm/sem.ZdMlaj") = 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("\xf8\x2f\x7d\x53\x9b\x6e\xd1\x79", 8, GRND\_NONBLOCK) = 8

newfstatat(AT\_FDCWD, "/dev/shm/sem.kLCN4s", 0x7ffcac843760, AT\_SYMLINK\_NOFOLLOW) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/dev/shm/sem.kLCN4s", O\_RDWR|O\_CREAT|O\_EXCL|O\_NOFOLLOW|O\_CLOEXEC, 0666) = 3

write(3, "\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\0", 32) = 32

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f3c6860f000

link("/dev/shm/sem.kLCN4s", "/dev/shm/sem.sem\_read2") = 0

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=32, ...}) = 0

unlink("/dev/shm/sem.kLCN4s") = 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("\x31\x0e\x26\xa4\xbb\xde\xba\x79", 8, GRND\_NONBLOCK) = 8

newfstatat(AT\_FDCWD, "/dev/shm/sem.RtnVNS", 0x7ffcac843760, AT\_SYMLINK\_NOFOLLOW) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/dev/shm/sem.RtnVNS", O\_RDWR|O\_CREAT|O\_EXCL|O\_NOFOLLOW|O\_CLOEXEC, 0666) = 3

write(3, "\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\0", 32) = 32

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f3c6860e000

link("/dev/shm/sem.RtnVNS", "/dev/shm/sem.sem\_write") = 0

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=32, ...}) = 0

unlink("/dev/shm/sem.RtnVNS") = 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) = 0x7f3c6860d000

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

read(0, output1.txt

"\321output1.txt\n", 1024) = 13

read(0, output2.txt

"output2.txt\n", 1024) = 12

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3c683f6a10) = 724

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3c683f6a10) = 725

read(0, aaaaaaa

"aaaaaaa\n", 1024) = 8

futex(0x7f3c68610000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860e000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY) = -1 EAGAIN (Resource temporarily unavailable)

read(0, jjjjjjj

"jjjjjjj\n", 1024) = 8

futex(0x7f3c68610000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860e000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

read(0, apchihba

"apchihba\n", 1024) = 9

futex(0x7f3c6860f000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860e000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

read(0, ahahahahah

"ahahahahah\n", 1024) = 11

futex(0x7f3c6860f000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860e000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY) = -1 EAGAIN (Resource temporarily unavailable)

read(0, skibidi dop dop

"skibidi dop dop\n", 1024) = 16

futex(0x7f3c6860f000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860e000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

read(0, yes yes yes

"yes yes yes\n", 1024) = 12

futex(0x7f3c68610000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860e000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

read(0, boldeagle

"boldeagle\n", 1024) = 10

futex(0x7f3c68610000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860e000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

read(0, q

"q\n", 1024) = 2

futex(0x7f3c68610000, FUTEX\_WAKE, 1) = 1

futex(0x7f3c6860f000, FUTEX\_WAKE, 1) = 1

wait4(-1, NULL, 0, NULL) = 724

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=724, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

wait4(-1, NULL, 0, NULL) = 725

munmap(0x7f3c68610000, 32) = 0

unlink("/dev/shm/sem.sem\_read1") = 0

munmap(0x7f3c6860f000, 32) = 0

unlink("/dev/shm/sem.sem\_read2") = 0

munmap(0x7f3c6860e000, 32) = 0

unlink("/dev/shm/sem.sem\_write") = 0

munmap(0x7f3c6860d000, 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.