

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Компьютерные науки и прикладная математика»  
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №5  
по курсу «Операционные системы»**

Выполнил: А. В. Маркелов  
Группа: М8О-207БВ-24  
Преподаватель: Е. С. Миронов

Москва, 2025

## Условие

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

**Задание:** При выполнении лабораторных работ по курсу ОС 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 ЛР должен содержать краткую сводку по исследованию написанных программ.

**Вариант:** -

## Метод решения

Реализованные к каждой лабораторной работе программы запускаются со strace логированием. Программам подаются тестовые данные, ожидается завершение выполнения и программа корректно завершается. Логи сохраняются в указанный при запуске файл формата .log.

## Описание программы

Ключевые строки strace логов к лабораторной работе 1:

```
1 21657 22:32:18.639973 pipe2([5,6],0) = 0
2 21657 22:32:18.640291 fcntl(5,F_SETFD,FD_CLOEXEC) = 0
3 21657 22:32:18.640585 fcntl(6,F_SETFD,FD_CLOEXEC) = 0
```

Создание двухнаправленного канала для связи родителя с дочерним процессом (pipe1 для нечётных строк). Флаг FD\_CLOEXEC автоматически закроет дескриптор при execve в дочернем процессе, предотвращая утечку файловых дескрипторов. Дескриптор 5 - конец чтения, 6 - конец записи.

```
1 21657 22:32:18.641945 clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|
    CLONE_CHILD_SETTID|SIGCHLD,child_tidptr=0x7f18563aea10) = 21767
2 21657 22:32:18.642471 clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|
    CLONE_CHILD_SETTID|SIGCHLD) = 21768
```

Создание двух независимых процессов. Флаг SIGCHLD отправляет сигнал родителю при завершении дочернего процесса. Каждый процесс получает уникальный PID и независимое адресное пространство. Это критично для изоляции дочерних процессов друг от друга.

```
1 21767 22:32:18.642693 dup2(5,0) = 0
2 21767 22:32:18.642998 dup2(3,1) = 1
```

Дочерний процесс перенаправляет stdin на конец чтения pipe (теперь читает данные от родителя) и stdout на дескриптор файла 3 (выходной файл "odd.txt"). Это создаёт четкую цепочку данных: parent -> pipe -> child stdin -> Reverse() -> child stdout -> file.

```
1 21767 22:32:18.643599 execve("./child",["child"],0x7fff95509278 /* 37 vars */) =
    0
```

Значение: Замена образа процесса 21767 на программу ./child. Перенаправленные дескрипторы (5→0, 3→1) остаются активными в новой программе. Это позволяет child.c читать из pipe и писать в файл без явного открытия этих ресурсов.

```
1 || 21657 waitpid(21767, &status, 0) = 21767
2 || 21657 waitpid(21768, &status, 0) = 21768
```

Родитель блокируется, ожидая завершения дочерних процессов. Это синхронизирует выполнение: родитель может только продолжить после полного завершения обработки всех данных дочерними процессами. В strace логах это проявляется как (unfinished) состояние.

Ключевые строки strace логов к лабораторной работе 2:

```
1 || 45186 rt_sigaction(SIGRT_1,{sa_handler=0x7fbc14281870,sa_mask=[],sa_flags=
    SA_RESTORER|SA_ON}) = 0
2 || 45186 rt_sigprocmask(SIG_UNBLOCK,[RTMIN RT_1],NULL,8) = 0
```

Регистрация обработчика real-time сигнала SIGRT\_1. Это необходимо для правильной работы механизма работы потоков в glibc. SIG\_UNBLOCK разрешает обработку сигнала в главном потоке.

```
1 || 45186 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0
    x7fbc139ec000
2 || 45186 mprotect(0x7fbc139ed000,8388608,PROT_READ|PROT_WRITE) = 0
```

Выделение 8 МБ памяти для стека каждого потока. Первый mmap с флагом PROT\_NONE создаёт защищённую область, затем mprotect устанавливает права на основную часть стека. Это позволяет ядру обнаружить переполнение стека (access на защищённую страницу → segfault).

```
1 || 45186 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
    CLONE_SYSVSEM|...},=>{parent_tid=[45187]},88) = 45187
```

Создание потока (не процесса). Флаг CLONE\_THREAD указывает, что это поток одного процесса. CLONE\_VM означает общее адресное пространство - все потоки видят одну память. CLONE\_SIGHAND означает общие обработчики сигналов.

```
1 || 45186 futex(0x7fbc1440ca70,FUTEX_WAIT_PRIVATE,2,NULL) = 0
2 || 45187 futex(0x7fbc1440ca70,FUTEX_WAIT_PRIVATE,...)
3 || 45186 futex(0x7fbc1440ca70,FUTEX_WAKE_PRIVATE,1) = 1
```

futex - это быстрый механизм синхронизации в ядре Linux. Адрес 0x7fbc1440ca70 содержит мьютекс. FUTEX\_WAIT\_PRIVATE блокирует поток в ядре, FUTEX\_WAKE\_PRIVATE пробуждает ожидающие потоки. Это позволяет синхронизировать обновление глобальной переменной best\_triangle без race conditions.

```
1 || 45186 futex(0x7fbc131ea910,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,45189,NULL,
    FUTEX_BITSET_M) = 0
```

Главный поток ждёт завершения потока с TID=45189. FUTEX\_WAIT\_BITSET использует битовую маску для более гибкой синхронизации. Это эквивалент pthread\_join() на уровне системных вызовов.

Ключевые строки strace логов к лабораторной работе 3:

```
1 || 92290 openat(AT_FDCWD, "/dev/shm/ ipc_buffer", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|
    O_CLOEXEC, 0666) = 3
2 || 92290 ftruncate(3, 1032) = 0
3 || 92290 mmap(NULL, 1032, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fea9c141000
```

Создание общей памяти между всеми процессами (1032 байта). Все три процесса (родитель + 2 дочерних) видят одну и ту же физическую память, но в разных виртуальных адресах.

```
1 || 92290 openat(AT_FDCWD, "/dev/shm/sem.QVvzXy", O_RDWR|O_CREAT|O_EXCL, 0666) = 4
2 || 92290 write(4, "\\\0\\0\\0\\0...", 32) = 32
3 || 92290 link("/dev/shm/sem.QVvzXy", "/dev/shm/sem.sem_parent_write") = 0
```

Создание четырех семафоров:

- sem\_parent\_write (init=1) - семафор для записи родителя
- sem\_child1\_read (init=0) - ждёт дочерний 1
- sem\_child2\_read (init=0) - ждёт дочерний 2
- sem\_ack (init=0) - подтверждение обработки

```
1 || 92290 futex(0x7fea9c106000, FUTEX_WAKE, 1) = 1
2 || 92291 futex(0x7f6170aa6000, FUTEX_WAIT_BITSET...) = 0
3 || 92291 write(4, "txet\\n", 5) = 5
4 || 92291 futex(0x7f6170aa4000, FUTEX_WAKE, 1) = 1
```

Синхронизация через futex:

- Родитель пробуждает дочерний процесс
- Дочерний обрабатывает и пишет результат
- Дочерний пробуждает родителя для следующей строки

```
1 || 92290 unlink("/dev/shm/sem.sem_parent_write") = 0
2 || 92290 unlink("/dev/shm/sem.sem_child1_read") = 0
3 || 92290 unlink("/dev/shm/sem.sem_child2_read") = 0
4 || 92290 unlink("/dev/shm/sem.sem_ack") = 0
5 || 92290 unlink("/dev/shm/IPC_buffer") = 0
```

Очистка ресурсов. Критично удалить все IPC объекты, иначе они остаются в /dev/shm и блокируют будущие запуски.

Ключевые строки strace логов к лабораторной работе 4:

```
1 || 41198 execve("./bin/program1", ["../bin/program1"], ...) = 0
```

Program1 статически скомпилирована с liblight.so. Все функции встроены прямо в исполняемый файл.

```
1 || 41198 mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0
    x7ff2629d5000
2 || 41198 mmap(0x7ff2629fd000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
    MAP_DENYWRITE, 3, 0x28000) = 0x7ff2629fd000
```

Загрузка libc в память. liblight уже встроена в program1.

```
1 || 41577 openat(AT_FDCWD, "./lib/liblight.so", O_RDONLY|O_CLOEXEC) = 3
2 || 41577 mmap(NULL, 16448, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0
    x7f20aad58000
3 || 41577 mmap(0x7f20aad59000, 4096, PROT_READ|PROT_EXEC, ...) = 0x7f20aad59000
4 || 41577 mmap(0x7f20aad5a000, 4096, PROT_READ, ...) = 0x7f20aad5a000
5 || 41577 mmap(0x7f20aad5b000, 8192, PROT_READ|PROT_WRITE, ...) = 0x7f20aad5b000
```

Динамическая загрузка liblight.so (program2). Четыре mmap для разных секций SO:

- Read-only header (ELF metadata)
- Executable code (PROT\_EXEC)
- Read-only data (strings, constants)

- Read-write data (initialized globals, BSS)

Путь относительный ("./lib/liblight.so") разрешается через getcwd()

```
1 41577 openat(AT_FDCWD, "./lib/libhard.so", O_RDONLY|O_CLOEXEC) = 3
2 41577 mmap(NULL, 16464, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0
   x7f20aab24000
```

Динамическая загрузка libhard.so. Обе библиотеки загружены одновременно в память.  
Позволяет быстрое переключение без перезагрузки.

```
1 41577 read(0, "0\n", 1024) = 2
2 41577 write(1, "Switched to impl 2 (Hard)\n\n", 27) = 27
3
4 41577 read(0, "1 1 20\n", 1024) = 7
5 41577 write(1, "PrimeCount(1, 20) = 8 (sieve)\n", 30) = 30
6
7 41577 read(0, "2 27\n", 1024) = 5
8 41577 write(1, "Decimal: 27\n", 12) = 12
9 41577 write(1, "Ternary: 1000\n", 14) = 14
10
11 41577 read(0, "0\n", 1024) = 2
12 41577 write(1, "Switched to impl 1 (Light)\n\n", 28) = 28
13
14 41577 read(0, "2 27\n", 1024) = 5
15 41577 write(1, "Binary: 11011\n", 14) = 14
```

Переключение через изменение указателей функций, БЕЗ перекомпиляции и перезагрузки. Обе реализации остаются в памяти для быстрого доступа

## Результаты

Получены 5 файлов содержащие strace логи к лабораторным работам (к лабораторным работам 1-3 по одному файлу, к лабораторной работе 4 приложены 2 файла для каждой тестовой программы program1 и program2).

## Выводы

Цели лабораторной работы достигнуты. Я провел диагностику всех программ лабораторных работ посредством анализа strace логов.

Все четыре лабораторные работы полностью соответствуют требованиям заданий. Они демонстрируют эволюцию подходов к параллельному программированию в Linux: от классических pipes к многопоточности, к IPC с изоляцией, к модульности и гибкости. Strace логи подтверждают корректность реализации на уровне системных вызовов, синхронизации и управления ресурсами.

Лабораторная работа 1:

Главный процесс (PID 21657) создаёт два pipe и два дочерних процесса через clone(). Каждый дочерний перенаправляет stdin/stdout на pipe и выходной файл через dup2, затем загружается программой child через execve. Родитель маршрутизирует строки в соответствующий pipe по чётности номера, дочерние инвертируют и пишут в файлы. Синхронизация встроена в pipe (блокирующие read/write). Результаты корректны: нечётные строки в file1.txt, чётные в file2.txt, все инвертированы. Завершение через EOF, waitpid ожидает обоих дочерних, успешный exit.

## Лабораторная работа 2:

Главный процесс (PID 45186) инициализирует сигналы через `rt_sigaction`, выделяет стеки (8 МБ × 4) через `mmap`, создаёт четыре потока через `clone3` с флагом `CLONE_VM` (общая память) и `CLONE_THREAD`. Все потоки видят одну переменную `best_triangle`, используют `futex` для синхронизации. Каждый поток обрабатывает свой диапазон комбинаций точек, локально ищет максимум, захватывает мьютекс только перед обновлением глобального. Результат: "Максимальная площадь: 10.706307" с координатами. Синхронизация через `futex WAIT/WAKE`, главный ждёт завершения через `futex WAIT_BITSET`.

## Лабораторная работа 3:

Родительский процесс (PID 92290) создаёт shared memory (`/dev/shm/ ipc_buffer`, 1032 байта) через `shm_open` и `mmap(MAP_SHARED)`, затем четыре семафора через `openat/write/link`. Создаёт два дочерних процесса (`clone + execve`), которые открывают существующий shared memory и семафоры. Основной цикл: родитель читает строку, определяет чётность, пробуждает соответствующий дочерний через `futex WAKE`, дочерний просыпается, инвертирует строку функцией `InvertLine()`, пишет в файл, сигнализирует завершение через `futex WAKE`. Родитель ждёт подтверждения через `futex WAIT_BITSET`. При EOF устанавливает `shutdown_flag`, пробуждает обоих, дочерние проверяют флаг и завершают. Результаты: `file1.txt` содержит инвертированные нечётные (`text→txet, 123das→sad321`), `file2.txt` содержит инвертированные чётные (`asd→dsa, Test→tseT`). Очистка: `wait4` для обоих, `unlink()` всех семафоров и shared memory.

## Лабораторная работа 4 (статическая линковка):

Program1 загружается через `execve`, статически слинкована с `liblight.so` (выбор реализации на этапе компоновки). Загружает `libc` через несколько `mmap` для различных секций (код, данные, BSS). Выводит "Program 1 (Static Linking - Naive Implementation)" и ожидает команд. Команда "1 1 20" → PrimeCount наивный алгоритм (проверка делимости) → результат 8 простых чисел → вывод "(naive)". Команда "2 5" → `translate_to_base(base=2)` → "Binary: 101"(5 = 101). Команда "exit" → "Goodbye!" → `exit_group(0)`. Program1 зафиксирована на наивной реализации и двоичном переводе, переключение невозможно (выбор сделан при компиляции).

## Лабораторная работа 4 (динамическая загрузка):

Program2 загружается через `execve` (не слинкована заранее с библиотеками). Загружает `libc`, затем выполняет ключевые операции: открывает `"./lib/liblight.so"` и отображает через `mmap` с четырьмя секциями (`PROT_READ, PROT_EXEC, PROT_READ, PROT_READ|PROT_WRITE`), открывает `"./lib/libhard.so"` аналогично. Обе библиотеки загружены одновременно в разные адреса (`liblight: 0x7f20aad58000, libhard: 0x7f20aab24000`) для быстрого переключения. Выводит "Program 2 (Dynamic Loading - Switchable Implementations)". Команда "0" → меняет указатели функций с `liblight` на `libhard` → "Switched to impl 2 (Hard)". Команда "1 1 20" → `libhard PrimeCount(Sieve)` → 8 → вывод "(sieve)". Команда "2 27" → `translate_to_base(base=3, ternary)` → "Ternary: 1000"(27 = 1000). Команда "0" → обратно на `impl 1` → "Switched to impl 1 (Light)". Команда "2 27" → `translate_to_base(base=2, binary)` → "Binary: 11011"(27 = 11011). Команда "exit" → "Goodbye!" → `munmap` выгружает библиотеки. Преимущество: Переключение реализаций без перекомпиляции, обе SO в памяти для мгновенного доступа.

## Исходная программа

```
31 21657 22:32:05.259838 mprotect(0x7f185661c000,8192,PROT_READ) = 0
32 21657 22:32:05.260012 prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max
    =RLIM64_INFINITY}) = 0
33 21657 22:32:05.260169 munmap(0x7f18565da000,29840) = 0
34 21657 22:32:05.260348 newfstatat(1,"",{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88
    ,0x8),...},AT_EMPTY_PATH) = 0
35 21657 22:32:05.260609 getrandom("\x62\x52\x12\xec\x54\xee\x63\xeb",8,
    GRND_NONBLOCK) = 8
36 21657 22:32:05.260871 brk(NULL) = 0x5577b2227000
37 21657 22:32:05.261210 brk(0x5577b2248000) = 0x5577b2248000
38 21657 22:32:05.261440 newfstatat(0,"",{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88
    ,0x8),...},AT_EMPTY_PATH) = 0
39 21657 22:32:05.261760 write(1,"Enter a file name for odd\nlines:",...,33) = 33
40 21657 22:32:05.262059 read(0,"odd\n",1024) = 4
41 21657 22:32:07.626601 write(1,"Enter a file name for even\nlines",...,34) = 34
42 21657 22:32:07.627084 read(0,"even\n",1024) = 5
43 21657 22:32:18.638922 openat(AT_FDCWD,"odd",O_WRONLY|O_CREAT|O_TRUNC,0666) = 3
44 21657 22:32:18.639383 openat(AT_FDCWD,"even",O_WRONLY|O_CREAT|O_TRUNC,0666) = 4
45 21657 22:32:18.639973 pipe2([5,6],0) = 0
46 21657 22:32:18.640291 fcntl(5,F_SETFD,FD_CLOEXEC) = 0
47 21657 22:32:18.640585 fcntl(6,F_SETFD,FD_CLOEXEC) = 0
48 21657 22:32:18.640937 pipe2([7,8],0) = 0
49 21657 22:32:18.641371 fcntl(7,F_SETFD,FD_CLOEXEC) = 0
50 21657 22:32:18.641663 fcntl(8,F_SETFD,FD_CLOEXEC) = 0
51 21657 22:32:18.641945 clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|
    CLONE_CHILD_SETTID|SIGCHLD,child_tidptr=0x7f18563aea10) = 21767
52 21767 22:32:18.642420 set_robust_list(0x7f18563aea20,24 <unfinished ...>
53 21657 22:32:18.642471 clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|
    CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>
54 21767 22:32:18.642517 <... set_robust_list resumed>) = 0
55 21767 22:32:18.642693 dup2(5,0) = 0
56 21657 22:32:18.642862 <... clone resumed>,child_tidptr=0x7f18563aea10) = 21768
57 21768 22:32:18.642955 set_robust_list(0x7f18563aea20,24 <unfinished ...>
58 21767 22:32:18.642998 dup2(3,1 <unfinished ...>
59 21657 22:32:18.643039 read(0, <unfinished ...>
60 21768 22:32:18.643101 <... set_robust_list resumed>) = 0
61 21767 22:32:18.643143 <... dup2 resumed>) = 1
62 21768 22:32:18.643185 dup2(7,0 <unfinished ...>
63 21767 22:32:18.643226 fcntl(5,F_SETFD,0 <unfinished ...>
64 21768 22:32:18.643270 <... dup2 resumed>) = 0
65 21767 22:32:18.643328 <... fcntl resumed>) = 0
66 21768 22:32:18.643370 dup2(4,1 <unfinished ...>
67 21767 22:32:18.643411 fcntl(3,F_SETFD,0 <unfinished ...>
68 21768 22:32:18.643453 <... dup2 resumed>) = 1
69 21767 22:32:18.643516 <... fcntl resumed>) = 0
70 21768 22:32:18.643558 fcntl(7,F_SETFD,0 <unfinished ...>
71 21767 22:32:18.643599 execve("./child",["child"],0x7fff95509278 /* 37 vars */ <
    unfinished ...>
72 21768 22:32:18.643685 <... fcntl resumed>) = 0
73 21768 22:32:18.643803 fcntl(4,F_SETFD,0) = 0
74 21767 22:32:18.644020 <... execve resumed>) = 0
```

```
75 21768 22:32:18.644064 execve("./child", ["child"], 0x7fff95509278 /* 37 vars */ <
    unfinished ...>
76 21767 22:32:18.644146 brk(NULL) = 0x55d3e77d4000
77 21768 22:32:18.644427 <... execve resumed>) = 0
78 21767 22:32:18.644498 arch_prctl(0x3001 /* ARCH_??? */, 0x7ffdde7f63110 <
    unfinished ...>
79 21768 22:32:18.644545 brk(NULL <unfinished ...>
80 21767 22:32:18.644586 <... arch_prctl resumed>) = -1 EINVAL (Invalid argument)
81 21768 22:32:18.644635 <... brk resumed>) = 0x55fb5508b000
82 21767 22:32:18.644723 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
    MAP_ANONYMOUS, -1, 0 <unfinished ...>
83 21768 22:32:18.644790 arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd114f7df0 <
    unfinished ...>
84 21767 22:32:18.644831 <... mmap resumed>) = 0x7f8eb9278000
85 21768 22:32:18.644873 <... arch_prctl resumed>) = -1 EINVAL (Invalid argument)
86 21767 22:32:18.644939 access("/etc/ld.so.preload", R_OK <unfinished ...>
87 21768 22:32:18.644990 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
    MAP_ANONYMOUS, -1, 0 <unfinished ...>
88 21767 22:32:18.645055 <... access resumed>) = -1 ENOENT (No such file or
    directory)
89 21768 22:32:18.645098 <... mmap resumed>) = 0x7f1dea5f1000
90 21767 22:32:18.645177 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <
    unfinished ...>
91 21768 22:32:18.645227 access("/etc/ld.so.preload", R_OK <unfinished ...>
92 21767 22:32:18.645273 <... openat resumed>) = 6
93 21768 22:32:18.645336 <... access resumed>) = -1 ENOENT (No such file or
    directory)
94 21767 22:32:18.645378 newfstatat(6, "", <unfinished ...>
95 21768 22:32:18.645423 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <
    unfinished ...>
96 21767 22:32:18.645491 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size
    =29840,...}, AT_EMPTY_PATH) = 0
97 21768 22:32:18.645539 <... openat resumed>) = 5
98 21767 22:32:18.645580 mmap(NULL, 29840, PROT_READ, MAP_PRIVATE, 6, 0 <unfinished ...>
99 21768 22:32:18.645643 newfstatat(5, "", <unfinished ...>
100 21767 22:32:18.645688 <... mmap resumed>) = 0x7f8eb9270000
101 21768 22:32:18.645728 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size
    =29840,...}, AT_EMPTY_PATH) = 0
102 21767 22:32:18.645800 close(6 <unfinished ...>
103 21768 22:32:18.645843 mmap(NULL, 29840, PROT_READ, MAP_PRIVATE, 5, 0 <unfinished ...>
104 21767 22:32:18.645912 <... close resumed>) = 0
105 21768 22:32:18.645952 <... mmap resumed>) = 0x7f1dea5e9000
106 21768 22:32:18.646105 close(5 <unfinished ...>
107 21767 22:32:18.646228 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY
    |O_CLOEXEC <unfinished ...>
108 21768 22:32:18.646324 <... close resumed>) = 0
109 21768 22:32:18.646554 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY
    |O_CLOEXEC <unfinished ...>
110 21767 22:32:18.646605 read(6, <unfinished ...>
111 21768 22:32:18.646672 <... openat resumed>) = 5
```

```
112 21767 22:32:18.646714 <... read resumed>"\\177ELF  
    "\\2\\1\\1\\3\\0\\0\\0\\0\\0\\0\\0\\3\\0>\\0\\1\\0\\0\\0P  
    \\237\\2\\0\\0\\0\\0\\0"...,832) = 832
```

### Листинг 1: \*Strace логи к лабораторной работе 1\*

```

29 45186 mmap(0x7fbc143ad000,360448,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE
   ,3,0x1bd000) = 0x7fbc143ad000
30 45186 mmap(0x7fbc14406000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|
   MAP_DENYWRITE,3) = 0x7fbc14406000
31 45186 mmap(0x7fbc1440c000,52816,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|
   MAP_ANONYMOUS,-1) = 0x7fbc1440c000
32 45186 close(3) = 0
33 45186 mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0
   x7fbc141ed000
34 45186 arch_prctl(ARCH_SET_FS,0x7fbc141ed740) = 0
35 45186 set_tid_address(0x7fbc141eda10) = 45186
36 45186 set_robust_list(0x7fbc141eda20,24) = 0
37 45186 rseq(0x7fbc141ee0e0,0x20,0,0x53053053) = 0
38 45186 mprotect(0x7fbc14406000,16384,PROT_READ) = 0
39 45186 mprotect(0x7fbc144fe000,4096,PROT_READ) = 0
40 45186 mprotect(0x55934e476000,4096,PROT_READ) = 0
41 45186 mprotect(0x7fbc14542000,8192,PROT_READ) = 0
42 45186 prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY
   }) = 0
43 45186 munmap(0x7fbc14500000,29840) = 0
44 45186 getrandom("\xa8\x7a\xf5\x0d\xc7\x47\xdf\x7e",8,GRND_NONBLOCK) = 8
45 45186 brk(NULL) = 0x55936a6b8000
46 45186 brk(0x55936a6d9000) = 0x55936a6d9000
47 45186 openat(AT_FDCWD,"./test_points.txt",O_RDONLY) = 3
48 45186 newfstatat(3,"",{st_mode=S_IFREG|0644,st_size=121,...},AT_EMPTY_PATH) = 0
49 45186 read(3,"0.0 0.0 0.0\n1.0 0.0 0.0\n0.0 1.0 "...,4096) = 121
50 45186 lseek(3,0,SEEK_SET) = 0
51 45186 read(3,"0.0 0.0 0.0\n1.0 0.0 0.0\n0.0 1.0 "...,4096) = 121
52 45186 read(3,"",4096) = 0
53 45186 lseek(3,0,SEEK_SET) = 0
54 45186 read(3,"0.0 0.0 0.0\n1.0 0.0 0.0\n0.0 1.0 "...,4096) = 121
55 45186 read(3,"",4096) = 0
56 45186 close(3) = 0
57 45186 newfstatat(1,"",{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0x5),...},
   AT_EMPTY_PATH) = 0
58 45186 write(1,
   "\\\320\\\227\\\320\\\260\\\320\\\263\\\321\\\200\\\321\\\203\\\320\\\266\\\320\\\265
   \\\320\\\275\\\320\\\276 10 \\\321\\\202\\\320\\\276\\\321\\\207\\\320\\\265\\\320\\\272"
   ...,33) = 33
59 45186 write(1,"\\\320\\\222\\\321\\\201\\\320\\\265\\\320\\\263\\\320\\\276
   \\\320\\\272\\\320\\\276\\\320\\\274\\\320\\\261\\\320\\\270\\\320
   \\\275\\\320\\\260\\\321\\\206\\\320\\\270\\\320\\\271":...,38) = 38
60 45186 rt_sigaction(SIGRT_1,{sa_handler=0x7fbc14281870,sa_mask=[],sa_flags=
   SA_RESTORER|SA_ON}) = 0
61 45186 rt_sigprocmask(SIG_UNBLOCK,[RTMIN RT_1],NULL,8) = 0
62 45186 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0
   x7fbc139ec000
63 45186 mprotect(0x7fbc139ed000,8388608,PROT_READ|PROT_WRITE) = 0
64 45186 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
65 45187 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
   CLONE_SYSVSEM|...},=>{parent_tid=[45187]},88) = 45187

```

```
68 45186 rt_sigprocmask(SIG_SETMASK, [],NULL,8) = 0
69 45186 write(1,"\\320\\237\\320\\276\\321\\202\\320\\276\\320\\272 0:
    \\320\\272\\320\\276\\320\\274\\320\\261\\320\\270\\320\\275\\320\\260\\321
70 \\206\\320\\270"...,58) = 58
71 45186 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0
    x7fbc131eb000
72 45186 mprotect(0x7fbc131ec000,8388608,PROT_READ|PROT_WRITE) = 0
73 45186 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
74 45188 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
    CLONE_SYSVSEM|...},=>{parent_tid=[45188]},88) = 45188
75 45186 rt_sigprocmask(SIG_SETMASK, [],NULL,8) = 0
76 45186 write(1,"\\320\\237\\320\\276\\321\\202\\320\\276\\320\\272 1:
    \\320\\272\\320\\276\\320\\274\\320\\261\\320\\270\\320\\275\\320\\260\\321
77 \\206\\320\\270"...,59) = 59
78 45186 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0
    x7fbc129ea000
79 45186 mprotect(0x7fbc129eb000,8388608,PROT_READ|PROT_WRITE) = 0
80 45186 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
81 45189 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
    CLONE_SYSVSEM|...},=>{parent_tid=[45189]},88) = 45189
82 45186 rt_sigprocmask(SIG_SETMASK, [],NULL,8) = 0
83 45186 futex(0x7fbc1440ca70,FUTEX_WAIT_PRIVATE,2,NULL) = 0
84 45186 write(1,"\\320\\237\\320\\276\\321\\202\\320\\276\\320\\272 2:
    \\320\\272\\320\\276\\320\\274\\320\\261\\320\\270\\320\\275\\320\\260\\321
85 \\206\\320\\270"...,59) = 59
86 45186 futex(0x7fbc1440ca70,FUTEX_WAKE_PRIVATE,1) = 1
87 45186 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0
    x7fbc121e9000
88 45186 mprotect(0x7fbc121ea000,8388608,PROT_READ|PROT_WRITE) = 0
89 45186 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
90 45190 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
    CLONE_SYSVSEM|...},=>{parent_tid=[45190]},88) = 45190
91 45186 rt_sigprocmask(SIG_SETMASK, [],NULL,8) = 0
92 45186 write(1,"\\320\\237\\320\\276\\321\\202\\320\\276\\320\\272 3:
    \\320\\272\\320\\276\\320\\274\\320\\261\\320\\270\\320\\275\\320\\260\\
93 321\\206\\320\\270"...,60) = 60
94 45186 futex(0x7fbc131ea910,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,45189,NULL,
    FUTEX_BITSET_M) = 0
95 45186 futex(0x7fbc129e9910,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,45190,NULL,
    FUTEX_BITSET_M) = 0
96 45186 write(1,"
    \\320\\240\\320\\225\\320\\227\\320\\243\\320\\233\\320\\254\\320\\242
97 \\320\\220\\320\\242\\320\\220\\320\\242\\320") = 21
98 45186 write(1,"
    \\320\\234\\320\\260\\320\\272\\321\\201\\320\\270\\320\\274\\320\\260
99 \\320\\273\\321\\214\\320 \\320\\277\\320\\273\\320\\276\\321"...,51) = 51
100 45186 write(1,"
    \\320\\222\\320\\265\\321\\200\\321\\210\\320\\270\\320\\275\\321\\213:\\\\n"
    ,16) = 16
101 45186 write(1,"P1: (1.500000,3.500000,1.00000"...,35) = 35
102 45186 write(1,"P2: (-1.000000,-1.000000,0.000"...,37) = 37
```

```
103 || 45186 write(1,"P3: (2.500000,0.500000,4.000000)...,35) = 35
104 || 45186 exit_group(0) = ?
105 || 45186 +++ exited with 0 +++
```

## Листинг 2: \*Strace логи к лабораторной работе 2\*









```

    unfinished ...>
184 92292 set_robust_list(0x7f7aacfd2a20, 24 <unfinished ...>
185 92291 <... mmap resumed> = 0x7f6170874000
186 92292 <... set_robust_list resumed> = 0
187 92292 rseq(0x7f7aacfd30e0, 0x20, 0, 0x53053053 <unfinished ...>
188 92291 arch_prctl(ARCH_SET_FS, 0x7f6170874740 <unfinished ...>
189 92292 <... rseq resumed> = 0
190 92291 <... arch_prctl resumed> = 0
191 92291 set_tid_address(0x7f6170874a10 <unfinished ...>
192 92292 mprotect(0x7f7aad1eb000, 16384, PROT_READ <unfinished ...>
193 92291 <... set_tid_address resumed> = 92291
194 92292 <... mprotect resumed> = 0
195 92291 set_robust_list(0x7f6170874a20, 24 <unfinished ...>
196 92292 mprotect(0x556bfe938000, 4096, PROT_READ <unfinished ...>
197 92291 <... set_robust_list resumed> = 0
198 92292 <... mprotect resumed> = 0
199 92291 rseq(0x7f61708750e0, 0x20, 0, 0x53053053 <unfinished ...>
200 92292 mprotect(0x7f7aad240000, 8192, PROT_READ <unfinished ...>
201 92291 <... rseq resumed> = 0
202 92292 <... mprotect resumed> = 0
203 92292 prlimit64(0, RLIMIT_STACK, NULL, <unfinished ...>
204 92291 mprotect(0x7f6170a8d000, 16384, PROT_READ <unfinished ...>
205 92292 <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY} = 0
206 92291 <... mprotect resumed> = 0
207 92292 munmap(0x7f7aad1fe000, 30076 <unfinished ...>
208 92291 mprotect(0x563128bee000, 4096, PROT_READ <unfinished ...>
209 92292 <... munmap resumed> = 0
210 92291 <... mprotect resumed> = 0
211 92292 openat(AT_FDCWD, "/dev/shm/ ipc_buffer", O_RDWR|O_NOFOLLOW|O_CLOEXEC <
    unfinished ...>
212 92291 mprotect(0x7f6170ae2000, 8192, PROT_READ <unfinished ...>
213 92292 <... openat resumed> = 3
214 92291 <... mprotect resumed> = 0
215 92292 mmap(NULL, 1032, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0 <unfinished ...>
216 92291 prlimit64(0, RLIMIT_STACK, NULL, <unfinished ...>
217 92292 <... mmap resumed> = 0x7f7aad23f000
218 92291 <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY} = 0
219 92292 openat(AT_FDCWD, "/dev/shm/ sem.sem_parent_write", O_RDWR|O_NOFOLLOW <
    unfinished ...>
220 92291 munmap(0x7f6170aa0000, 30076 <unfinished ...>
221 92292 <... openat resumed> = 4
222 92291 <... munmap resumed> = 0
223 92292 newfstatat(4, "", <unfinished ...>
224 92291 openat(AT_FDCWD, "/dev/shm/ ipc_buffer", O_RDWR|O_NOFOLLOW|O_CLOEXEC <
    unfinished ...>
225 92292 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=32, ...},
    AT_EMPTY_PATH) = 0
226 92291 <... openat resumed> = 3
227 92292 getrandom(<unfinished ...>
228 92291 mmap(NULL, 1032, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0 <unfinished ...>
229 92292 <... getrandom resumed>"\x56\xf5\xd5\x1f\x1c\x5f\xd2\x19", 8,
```

```
    GRND_NONBLOCK) = 8
230 92291 <... mmap resumed>          = 0x7f6170ae1000
231 92292 brk(NULL <unfinished ...>
232 92291 openat(AT_FDCWD, "/dev/shm/sem.sem_parent_write", O_RDWR|O_NOFOLLOW <
    unfinished ...>
233 92292 <... brk resumed>          = 0x556bff424000
234 92291 <... openat resumed>        = 4
235 92292 brk(0x556bff445000 <unfinished ...>
236 92291 newfstatat(4, "", <unfinished ...>
237 92292 <... brk resumed>          = 0x556bff445000
238 92291 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=32, ...},
    AT_EMPTY_PATH) = 0
239 92292 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>
240 92291 getrandom(<unfinished ...>
241 92292 <... mmap resumed>          = 0x7f7aad205000
242 92291 <... getrandom resumed>"\xea\xe6\x45\xf2\x62\x7b\x0c\xb2", 8,
    GRND_NONBLOCK) = 8
243 92292 close(4 <unfinished ...>
244 92291 brk(NULL <unfinished ...>
245 92292 <... close resumed>         = 0
246 92291 <... brk resumed>          = 0x56314f8fc000
247 92292 openat(AT_FDCWD, "/dev/shm/sem.sem_child1_read", O_RDWR|O_NOFOLLOW <
    unfinished ...>
248 92291 brk(0x56314f91d000 <unfinished ...>
249 92292 <... openat resumed>        = 4
250 92291 <... brk resumed>          = 0x56314f91d000
251 92292 newfstatat(4, "", <unfinished ...>
252 92291 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>
253 92292 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=32, ...},
    AT_EMPTY_PATH) = 0
254 92291 <... mmap resumed>          = 0x7f6170aa7000
255 92292 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>
256 92291 close(4 <unfinished ...>
257 92292 <... mmap resumed>          = 0x7f7aad204000
258 92291 <... close resumed>         = 0
259 92292 close(4 <unfinished ...>
260 92291 openat(AT_FDCWD, "/dev/shm/sem.sem_child1_read", O_RDWR|O_NOFOLLOW <
    unfinished ...>
261 92292 <... close resumed>         = 0
262 92292 openat(AT_FDCWD, "/dev/shm/sem.sem_child2_read", O_RDWR|O_NOFOLLOW <
    unfinished ...>
263 92291 <... openat resumed>        = 4
264 92292 <... openat resumed>        = 4
265 92291 newfstatat(4, "", <unfinished ...>
266 92292 newfstatat(4, "", <unfinished ...>
267 92291 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=32, ...},
    AT_EMPTY_PATH) = 0
268 92292 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=32, ...},
    AT_EMPTY_PATH) = 0
269 92291 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>
270 92292 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>
```

```
271 92291 <... mmap resumed> = 0x7f6170aa6000
272 92292 <... mmap resumed> = 0x7f7aad203000
273 92291 close(4 <unfinished ...>
274 92292 close(4 <unfinished ...>
275 92291 <... close resumed> = 0
276 92292 <... close resumed> = 0
277 92291 openat(AT_FDCWD, "/dev/shm/sem.sem_child2_read", O_RDWR|O_NOFOLLOW <
    unfinished ...>
278 92292 openat(AT_FDCWD, "/dev/shm/sem.sem_ack", O_RDWR|O_NOFOLLOW <unfinished
    ...>
279 92291 <... openat resumed> = 4
280 92292 <... openat resumed> = 4
281 92291 newfstatat(4, "", <unfinished ...>
282 92292 newfstatat(4, "", {st_mode=S_IFREG|0644, st_size=32, ...}, AT_EMPTY_PATH)
    = 0
283 92291 <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=32, ...},
    AT_EMPTY_PATH) = 0
284 92292 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>
285 92291 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>
286 92292 <... mmap resumed> = 0x7f7aad202000
287 92291 <... mmap resumed> = 0x7f6170aa5000
288 92292 close(4 <unfinished ...>
289 92291 close(4 <unfinished ...>
290 92292 <... close resumed> = 0
291 92291 <... close resumed> = 0
292 92292 openat(AT_FDCWD, "file2.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666 <unfinished
    ...>
293 92291 openat(AT_FDCWD, "/dev/shm/sem.sem_ack", O_RDWR|O_NOFOLLOW <unfinished
    ...>
294 92292 <... openat resumed> = 4
295 92291 <... openat resumed> = 4
296 92292 futex(0x7f7aad203000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
297 92291 newfstatat(4, "", {st_mode=S_IFREG|0644, st_size=32, ...}, AT_EMPTY_PATH)
    = 0
298 92291 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f6170aa4000
299 92291 close(4) = 0
300 92291 openat(AT_FDCWD, "file1.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4
301 92291 futex(0x7f6170aa6000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
302 92290 <... clock_nanosleep resumed>0x7fff28a95d50) = 0
303 92290 newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x9), ...},
    AT_EMPTY_PATH) = 0
304 92290 write(1, "Enter lines (empty line or Ctrl+..., 44) = 44
305 92290 newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x9), ...},
    AT_EMPTY_PATH) = 0
306 92290 read(0, "text\n", 1024) = 5
307 92290 futex(0x7fea9c106000, FUTEX_WAKE, 1) = 1
308 92291 <... futex resumed> = 0
309 92290 futex(0x7fea9c104000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
```

```
310 92291 write(4, "txet\n", 5) = 5
311 92291 futex(0x7f6170aa4000, FUTEX_WAKE, 1 <unfinished ...>
312 92290 <... futex resumed>) = 0
313 92291 <... futex resumed>) = 1
314 92290 read(0, <unfinished ...>
315 92291 futex(0x7f6170aa6000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
316 92290 <... read resumed>"asd\n", 1024) = 4
317 92290 futex(0x7fea9c105000, FUTEX_WAKE, 1) = 1
318 92292 <... futex resumed>) = 0
319 92290 futex(0x7fea9c104000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
320 92292 write(4, "dsa\n", 4) = 4
321 92292 futex(0x7f7aad202000, FUTEX_WAKE, 1 <unfinished ...>
322 92290 <... futex resumed>) = 0
323 92292 <... futex resumed>) = 1
324 92290 read(0, <unfinished ...>
325 92292 futex(0x7f7aad203000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
326 92290 <... read resumed>"123das\n", 1024) = 7
327 92290 futex(0x7fea9c106000, FUTEX_WAKE, 1) = 1
328 92291 <... futex resumed>) = 0
329 92290 futex(0x7fea9c104000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
330 92291 write(4, "sad321\n", 7) = 7
331 92291 futex(0x7f6170aa4000, FUTEX_WAKE, 1 <unfinished ...>
332 92290 <... futex resumed>) = 0
333 92291 <... futex resumed>) = 1
334 92290 read(0, <unfinished ...>
335 92291 futex(0x7f6170aa6000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
336 92290 <... read resumed>"Test\n", 1024) = 5
337 92290 futex(0x7fea9c105000, FUTEX_WAKE, 1) = 1
338 92292 <... futex resumed>) = 0
339 92290 futex(0x7fea9c104000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
340 92292 write(4, "tseT\n", 5) = 5
341 92292 futex(0x7f7aad202000, FUTEX_WAKE, 1 <unfinished ...>
342 92290 <... futex resumed>) = 0
343 92292 <... futex resumed>) = 1
344 92290 read(0, <unfinished ...>
345 92292 futex(0x7f7aad203000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL,
    FUTEX_BITSET_MATCH_ANY <unfinished ...>
346 92290 <... read resumed>"\n", 1024) = 0
347 92290 futex(0x7fea9c106000, FUTEX_WAKE, 1) = 1
348 92291 <... futex resumed>) = 0
349 92290 futex(0x7fea9c105000, FUTEX_WAKE, 1 <unfinished ...>
350 92291 close(4 <unfinished ...>
351 92290 <... futex resumed>) = 1
352 92292 <... futex resumed>) = 0
353 92290 wait4(92291, <unfinished ...>
```

```
354 || 92292 close(4 <unfinished ...>
355 92291 <... close resumed>) = 0
356 92292 <... close resumed>) = 0
357 92291 munmap(0x7f6170aa7000, 32 <unfinished ...>
358 92292 munmap(0x7f7aad205000, 32 <unfinished ...>
359 92291 <... munmap resumed>) = 0
360 92292 <... munmap resumed>) = 0
361 92291 munmap(0x7f6170aa6000, 32 <unfinished ...>
362 92292 munmap(0x7f7aad204000, 32 <unfinished ...>
363 92291 <... munmap resumed>) = 0
364 92292 <... munmap resumed>) = 0
365 92291 munmap(0x7f6170aa5000, 32 <unfinished ...>
366 92292 munmap(0x7f7aad203000, 32 <unfinished ...>
367 92291 <... munmap resumed>) = 0
368 92292 <... munmap resumed>) = 0
369 92291 munmap(0x7f6170aa4000, 32 <unfinished ...>
370 92292 munmap(0x7f7aad202000, 32 <unfinished ...>
371 92291 <... munmap resumed>) = 0
372 92292 <... munmap resumed>) = 0
373 92291 munmap(0x7f6170ae1000, 1032 <unfinished ...>
374 92292 munmap(0x7f7aad23f000, 1032 <unfinished ...>
375 92291 <... munmap resumed>) = 0
376 92292 <... munmap resumed>) = 0
377 92291 close(3 <unfinished ...>
378 92292 close(3 <unfinished ...>
379 92291 <... close resumed>) = 0
380 92292 <... close resumed>) = 0
381 92292 exit_group(0 <unfinished ...>
382 92291 exit_group(0 <unfinished ...>
383 92292 <... exit_group resumed>) = ?
384 92291 <... exit_group resumed>) = ?
385 92292 +++ exited with 0 ***
386 92291 +++ exited with 0 ***
387 92290 <... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) =
    92291
388 92290 --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=92292, si_uid
    =1000, si_status=0, si_utime=0, si_stime=1} ---
389 92290 wait4(92292, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 92292
390 92290 munmap(0x7fea9c107000, 32) = 0
391 92290 munmap(0x7fea9c106000, 32) = 0
392 92290 munmap(0x7fea9c105000, 32) = 0
393 92290 munmap(0x7fea9c104000, 32) = 0
394 92290 munmap(0x7fea9c141000, 1032) = 0
395 92290 close(3) = 0
396 92290 unlink("/dev/shm/sem.sem_parent_write") = 0
397 92290 unlink("/dev/shm/sem.sem_child1_read") = 0
398 92290 unlink("/dev/shm/sem.sem_child2_read") = 0
399 92290 unlink("/dev/shm/sem.sem_ack") = 0
400 92290 unlink("/dev/shm/ ipc_buffer") = 0
401 92290 exit_group(0) = ?
```

402 || 92290 +++ exited with 0 +++

### Листинг 3: \*Strace логи к лабораторной работе 3\*

```
34 newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x9), ...},  
35 AT_EMPTY_PATH) = 0  
36 getrandom("\x7e\x6d\x41\x89\x41\xd2\x19\x02", 8, GRND_NONBLOCK) = 8  
37 brk(NULL) = 0x55f1f9031000  
38 brk(0x55f1f9052000) = 0x55f1f9052000  
39 write(1, "\n", 1) = 1  
40 write(1, "==== Program 1 (Static Linking - "..., 58) = 58  
41 write(1, "Commands: 1 A B | 2 number | hel"..., 41) = 41  
42 write(1, "\n", 1) = 1  
43 newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x9), ...},  
44 AT_EMPTY_PATH) = 0  
45 read(0, "1 1 20\n", 1024) = 7  
46 write(1, "PrimeCount(1, 20) = 8 (naive)\n", 30) = 30  
47 read(0, "2 5\n", 1024) = 4  
48 write(1, "Decimal: 5\n", 11) = 11  
49 write(1, "Binary: 101\n", 12) = 12  
50 read(0, "exit\n", 1024) = 5  
51 write(1, "Goodbye!\n", 9) = 9  
52 exit_group(0) = ?  
53 +++ exited with 0 +++
```

Листинг 4: \*Strace логи к лабораторной работе 4 (program1)\*



```

    AT_EMPTY_PATH) = 0
58 write(1, "\n", 1) = 1
59 write(1, "==== Program 2 (Dynamic Loading -"..., 65) = 65
60 write(1, "Commands: 0 | 1 A B | 2 number |"..., 45) = 45
61 write(1, "\n", 1) = 1
62 newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x9), ...},
    AT_EMPTY_PATH) = 0
63 read(0, "0\n", 1024) = 2
64 write(1, "Switched to impl 2 (Hard)\n\n", 27) = 27
65 read(0, "1 1 20\n", 1024) = 7
66 write(1, "PrimeCount(1, 20) = 8 (sieve)\n", 30) = 30
67 read(0, "2 27\n", 1024) = 5
68 write(1, "Decimal: 27\n", 12) = 12
69 write(1, "Ternary: 1000\n", 14) = 14
70 read(0, "0\n", 1024) = 2
71 write(1, "Switched to impl 1 (Light)\n\n", 28) = 28
72 read(0, "2 27\n", 1024) = 5
73 write(1, "Decimal: 27\n", 12) = 12
74 write(1, "Binary: 11011\n", 14) = 14
75 read(0, "exit\n", 1024) = 5
76 munmap(0x7f20aad58000, 16448) = 0
77 munmap(0x7f20aab24000, 16464) = 0
78 write(1, "Goodbye!\n", 9) = 9
79 exit_group(0) = ?
80 +++ exited with 0 +++

```

Листинг 5: \*Strace логи к лабораторной работе 4 (program2)\*