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»**

Группа: М8О-215Б-23

Студент: Венгер И.В. Преподаватель: Миронов Е.С. Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2025

# Постановка задачи

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.

# Общий метод и алгоритм решения

Для демонстрации системных вызовов будем использовать утилиту strace в операционной системе linux.

Strace — это диагностическая, отладочная и обучающая утилита пользовательского пространства для Linux. Она используется для мониторинга и вмешательства во взаимодействия между процессами и ядром Linux, включая системные вызовы, доставку сигналов и изменение состояния процесса. Работа strace возможна благодаря функции ядра, известной как ptrace.

Наиболее часто он используется для запуска программы и вывода списка системных вызовов, которые она выполняет. Это особенно полезно, когда программа часто аварийно завершает работу или ведет себя непредсказуемо; например, strace может показать, что программа пытается получить доступ к несуществующему или недоступному файлу.

Еще одно полезное применение strace — это использование флага -p для присоединения к уже запущенному процессу. Это может быть полезно, если процесс перестал отвечать, и strace может помочь определить, что процесс заблокирован при попытке установить сетевое соединение.

Кроме того, strace предоставляет множество других возможностей:

* Фильтрация системных вызовов: Вы можете указать, какие системные вызовы следует отслеживать, используя параметр -e trace=. Это можно сделать по имени (например, clone, fork, vfork), с использованием предопределенных групп (например, %ipc или %file) или с помощью регулярных выражений (начиная с версии strace 4.17, например, -e trace=/clock\_.\*).
* Трассировка путей: Вы можете указать список путей для трассировки, используя параметр -P (например, -P /etc/ld.so.cache).
* Выгрузка ввода-вывода: Вы можете указать список файловых дескрипторов для выгрузки ввода-вывода, используя параметры -e read= и -e write=.
* Подсчет времени выполнения: Strace позволяет подсчитывать время выполнения системных вызовов и их количество, используя параметры -T, -c, -C и -w. Параметр -U позволяет выводить дополнительную информацию, такую как минимальное и максимальное время выполнения системного вызова.
* Временные метки: Strace может печатать относительные или абсолютные временные метки, используя опции -t и -r.
* Изменение системных вызовов: Strace позволяет изменять выполняемые системные вызовы, включая изменение возвращаемого значения и кода ошибки, внедрение сигналов, задержек и изменение данных, на которые указывают аргументы системных вызовов. Это достигается с помощью опции -e inject=syscall specification:tampering specification:retval=:error=:signal=:delay\_enter=:delay\_exit=:poke\_enter=:poke\_exit=.
* Информация о дескрипторах файлов: Strace может извлекать информацию о дескрипторах файлов, включая сокеты, используя опцию -y. Опция -yy предоставляет дополнительную информацию, такую как адреса конечных точек для сокетов, пути и основные/дополнительные номера устройств для файлов.
* Трассировка стека: Strace может печатать трассировки стека, включая разборку символов (начиная с версии strace 4.21, используя опцию -k).

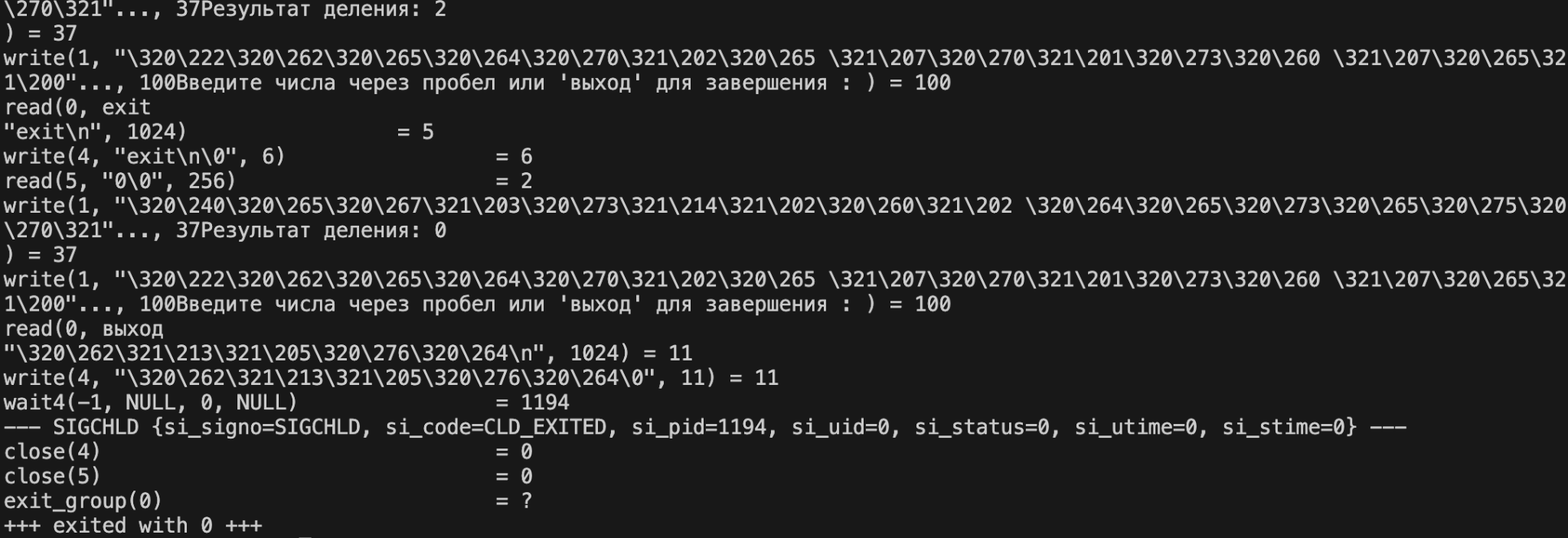
Strace также поддерживает множество других опций и параметров, которые делают его мощным инструментом для диагностики и отладки программного обеспечения.

# Протокол работы программы

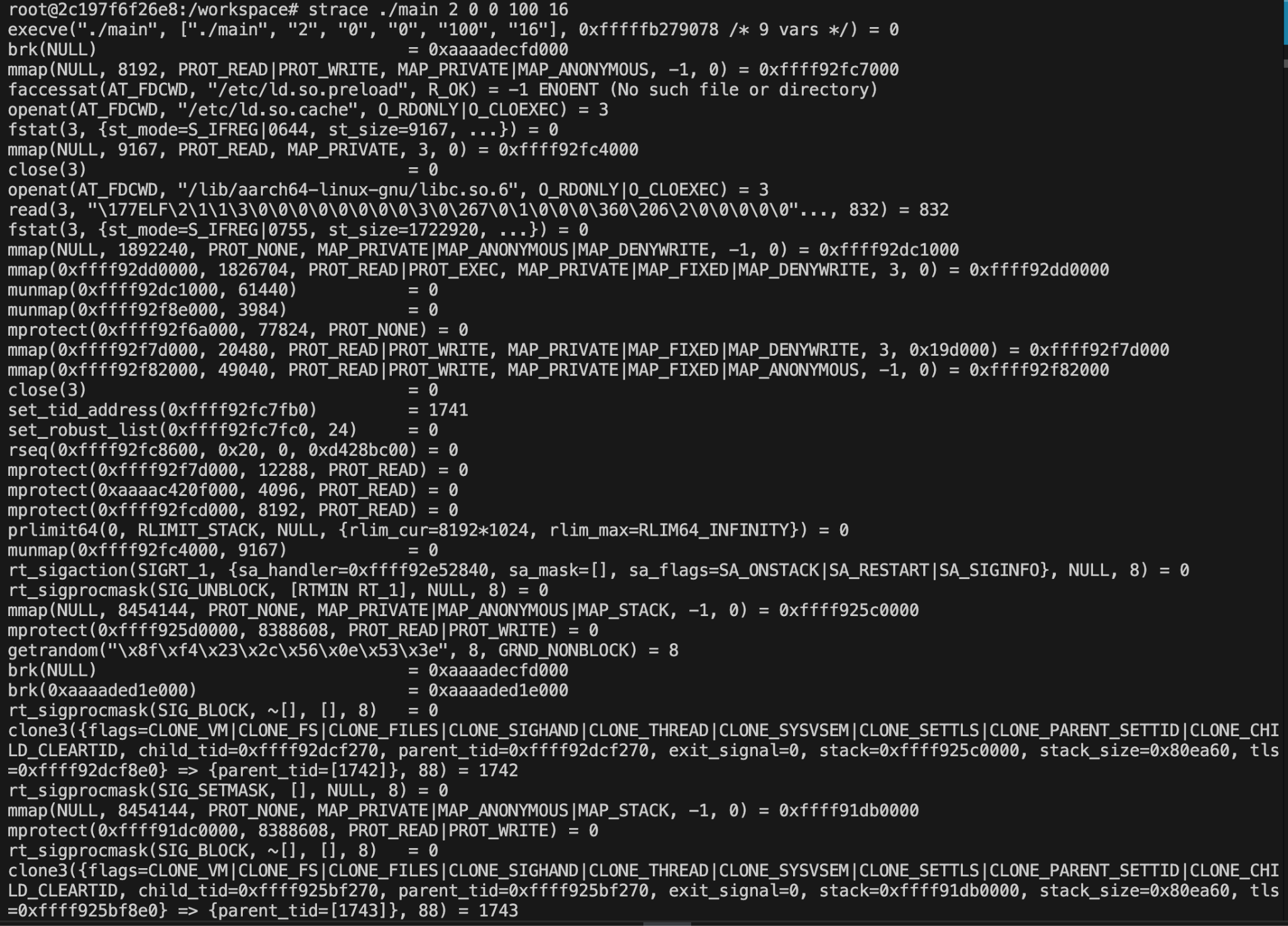
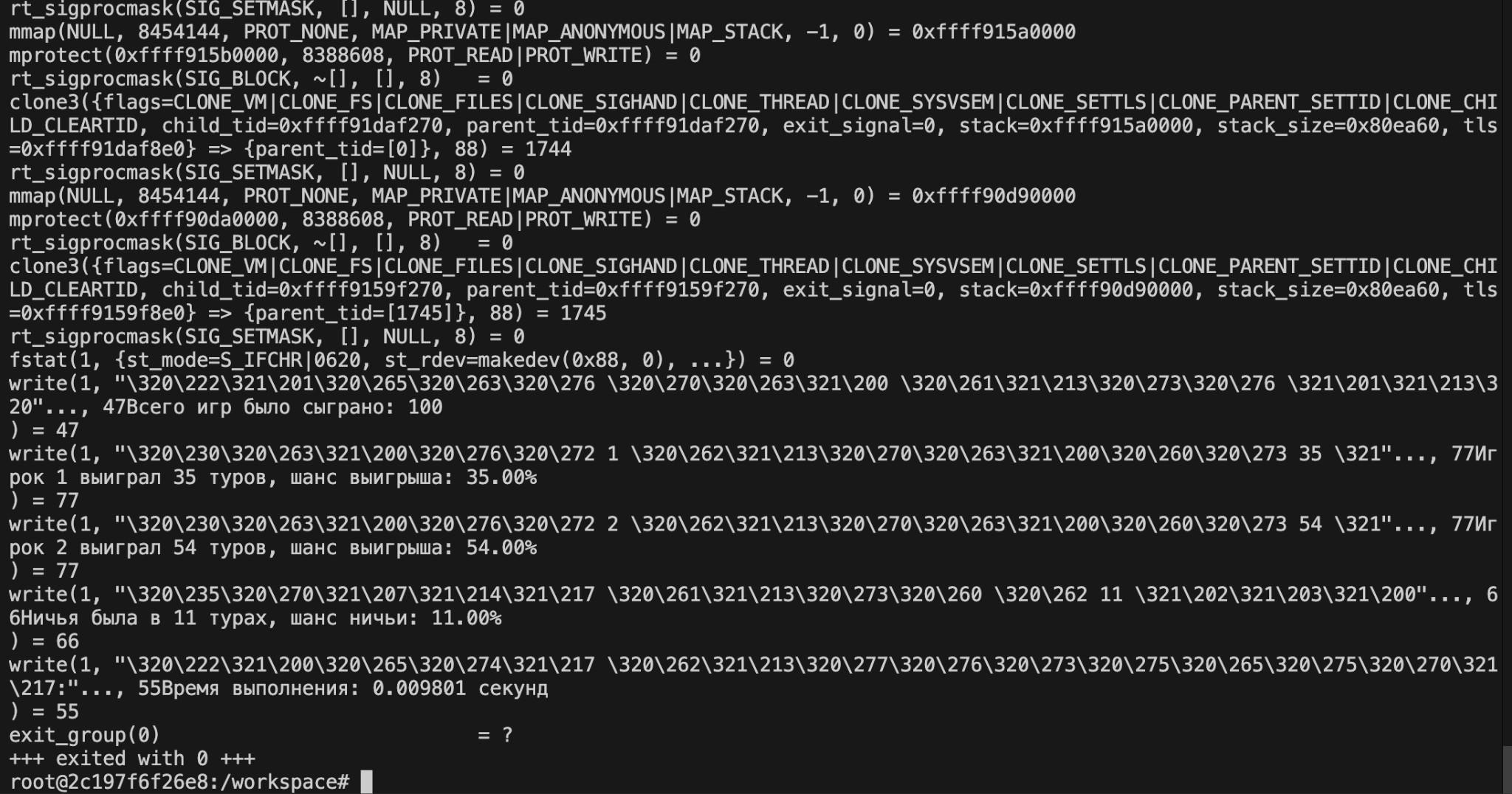
## Лабораторная работа №1.



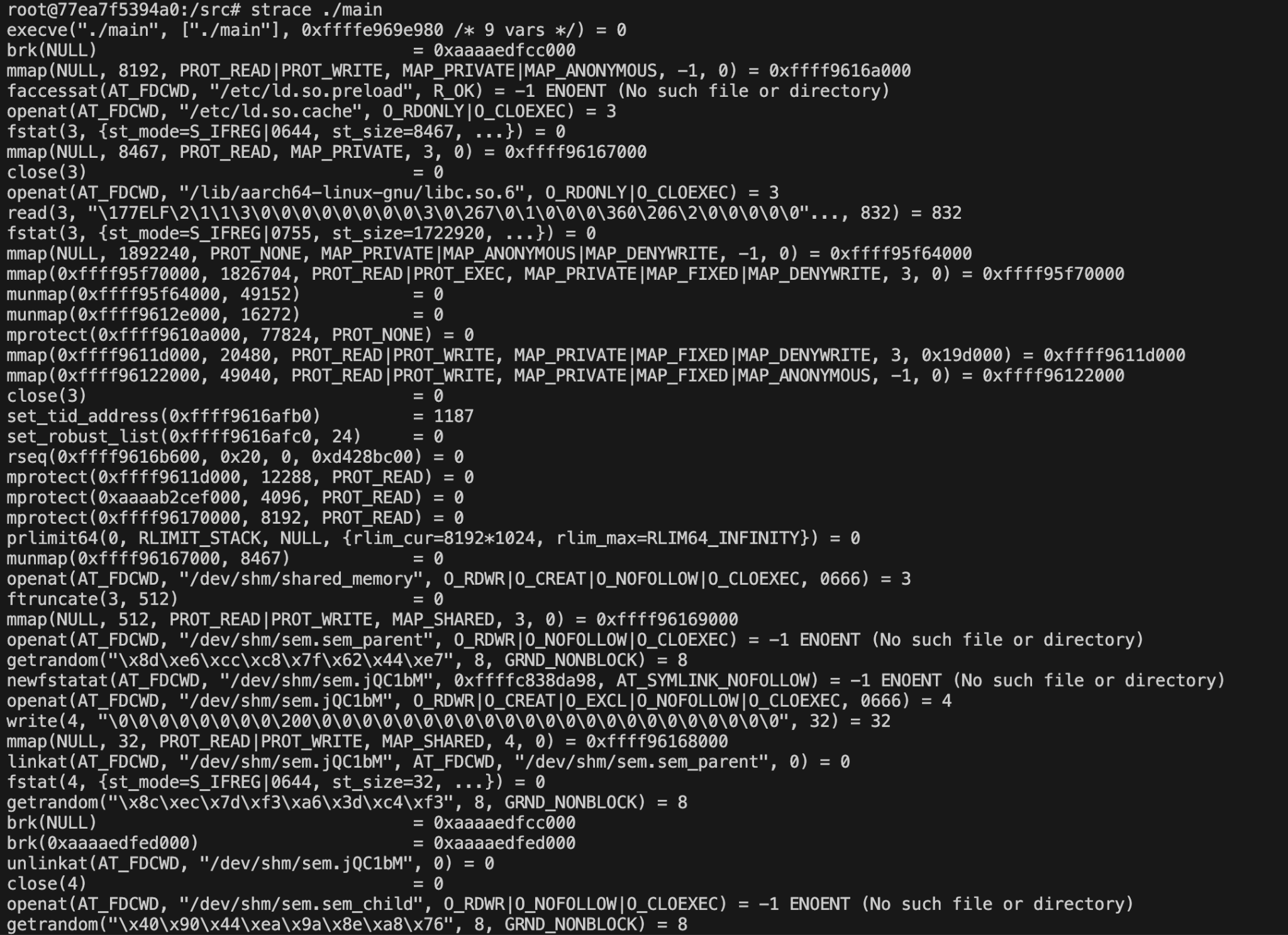




## Лабораторная работа №2.



## Лабораторная работа №3.

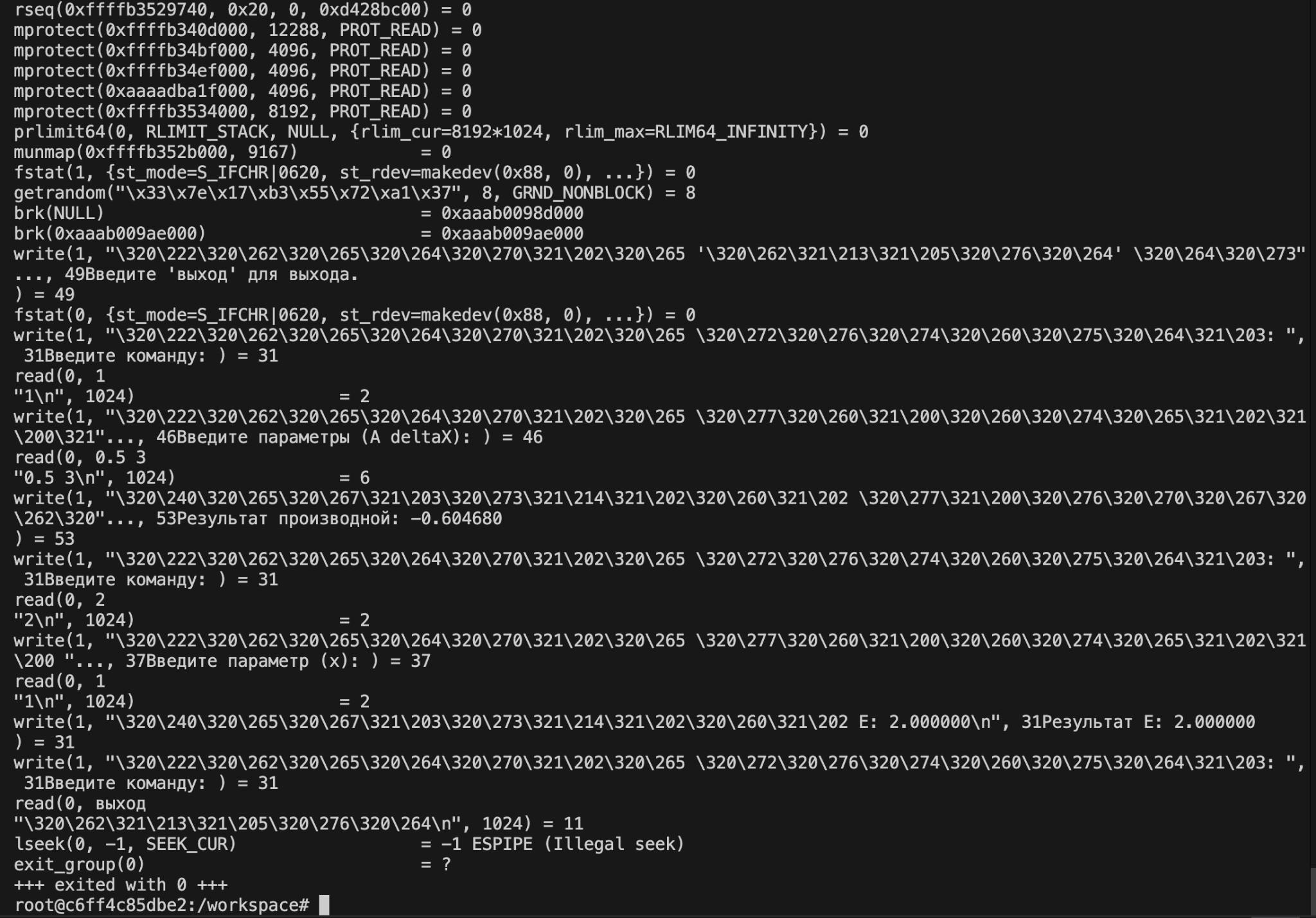
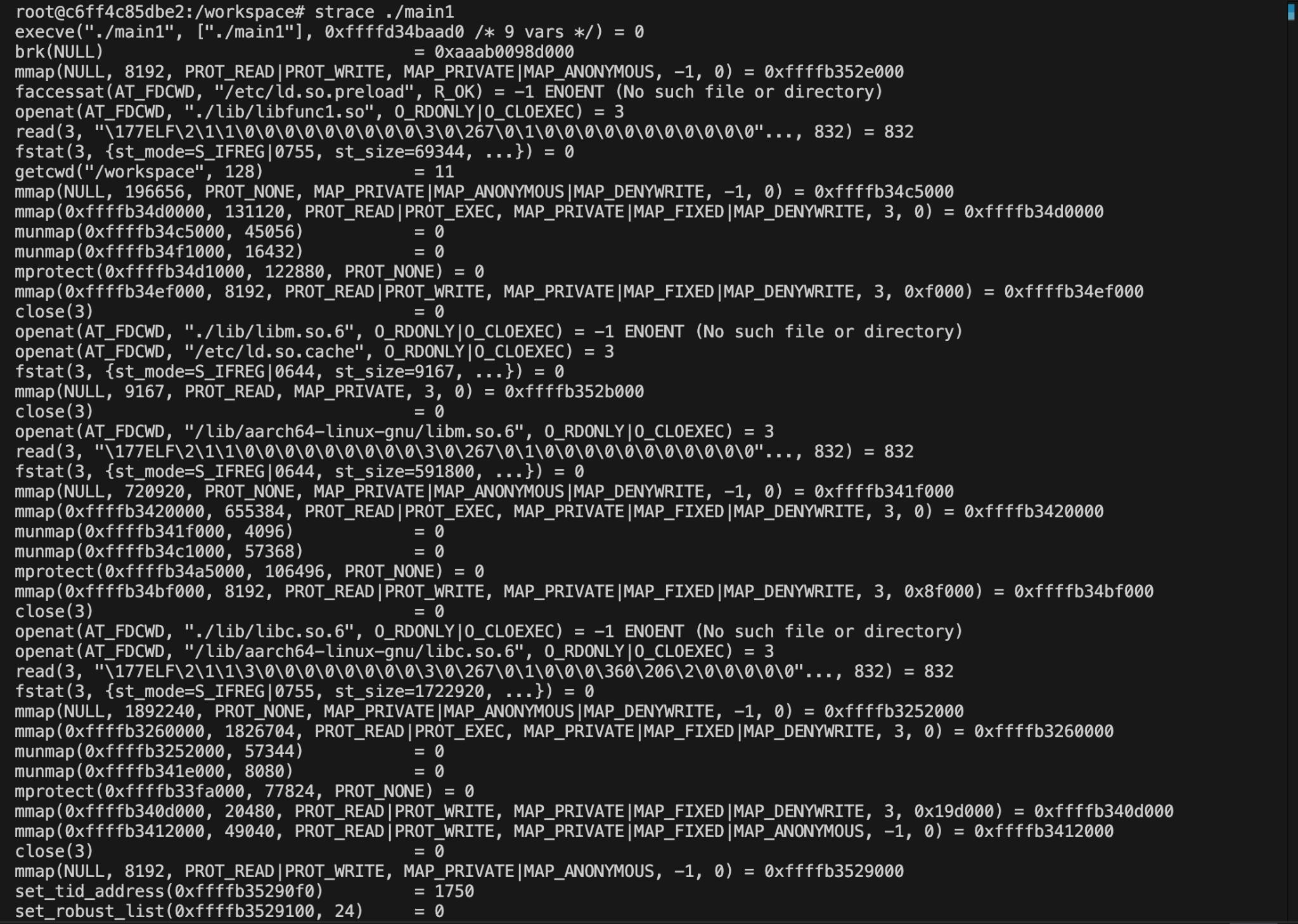




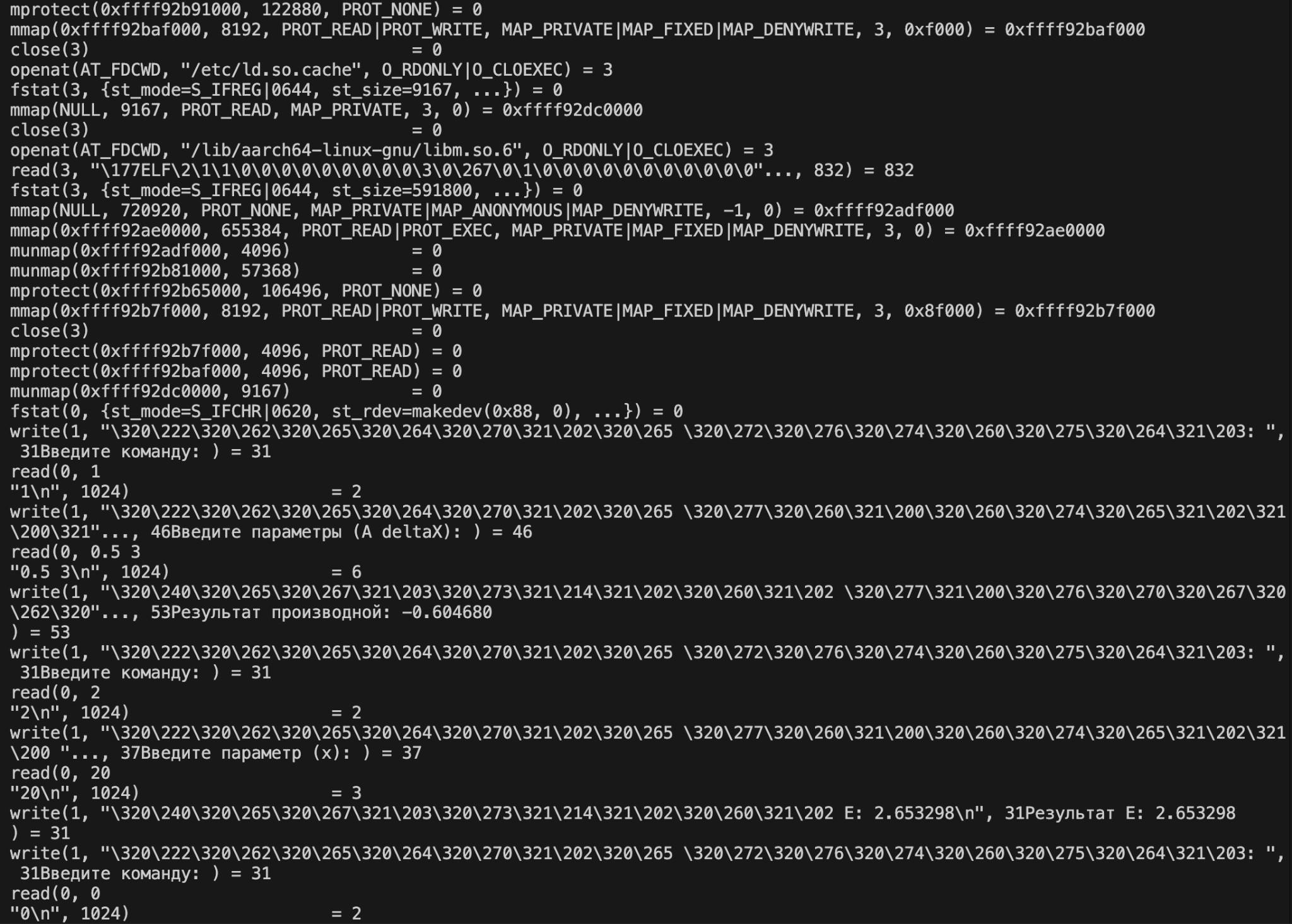
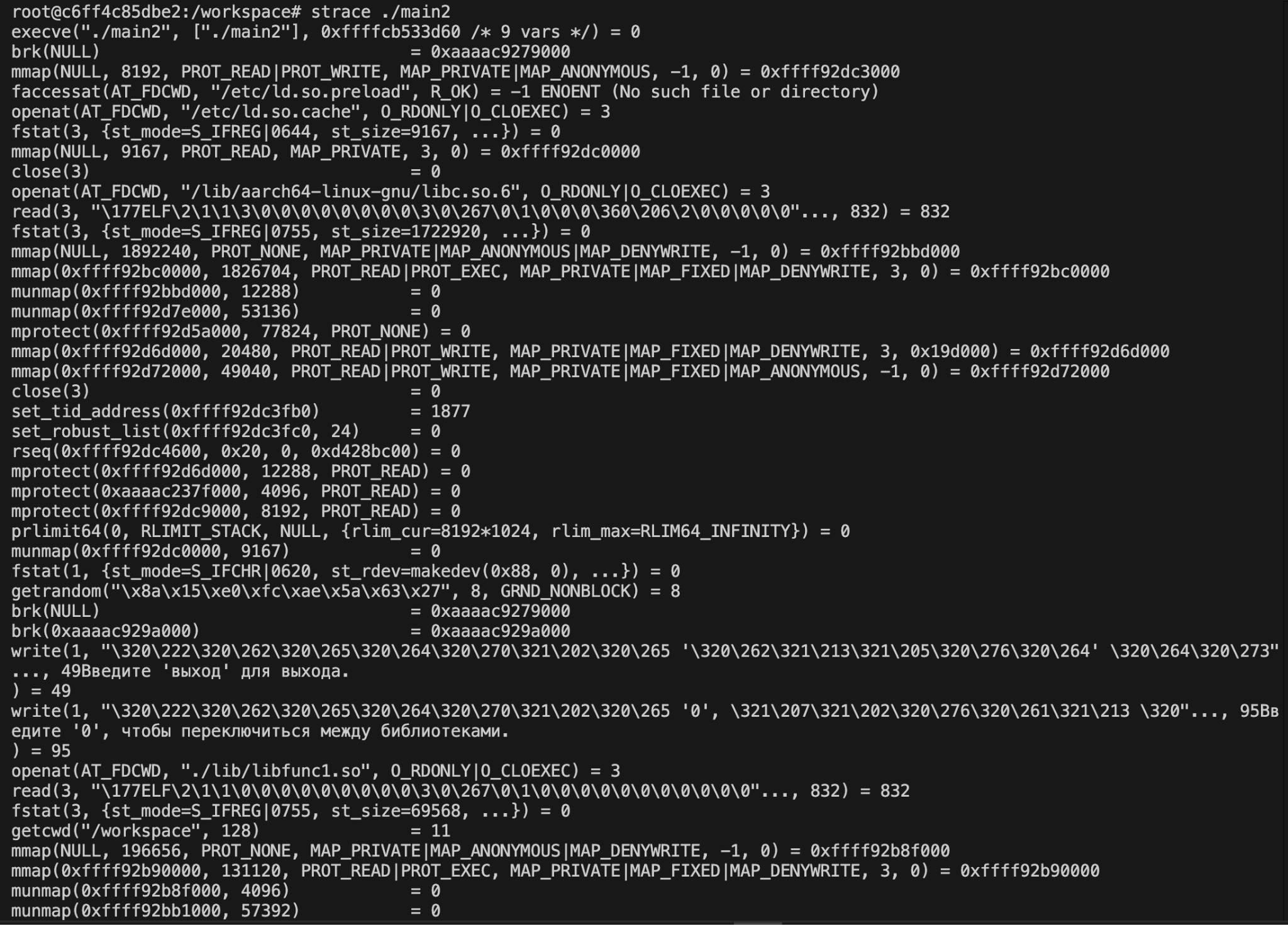


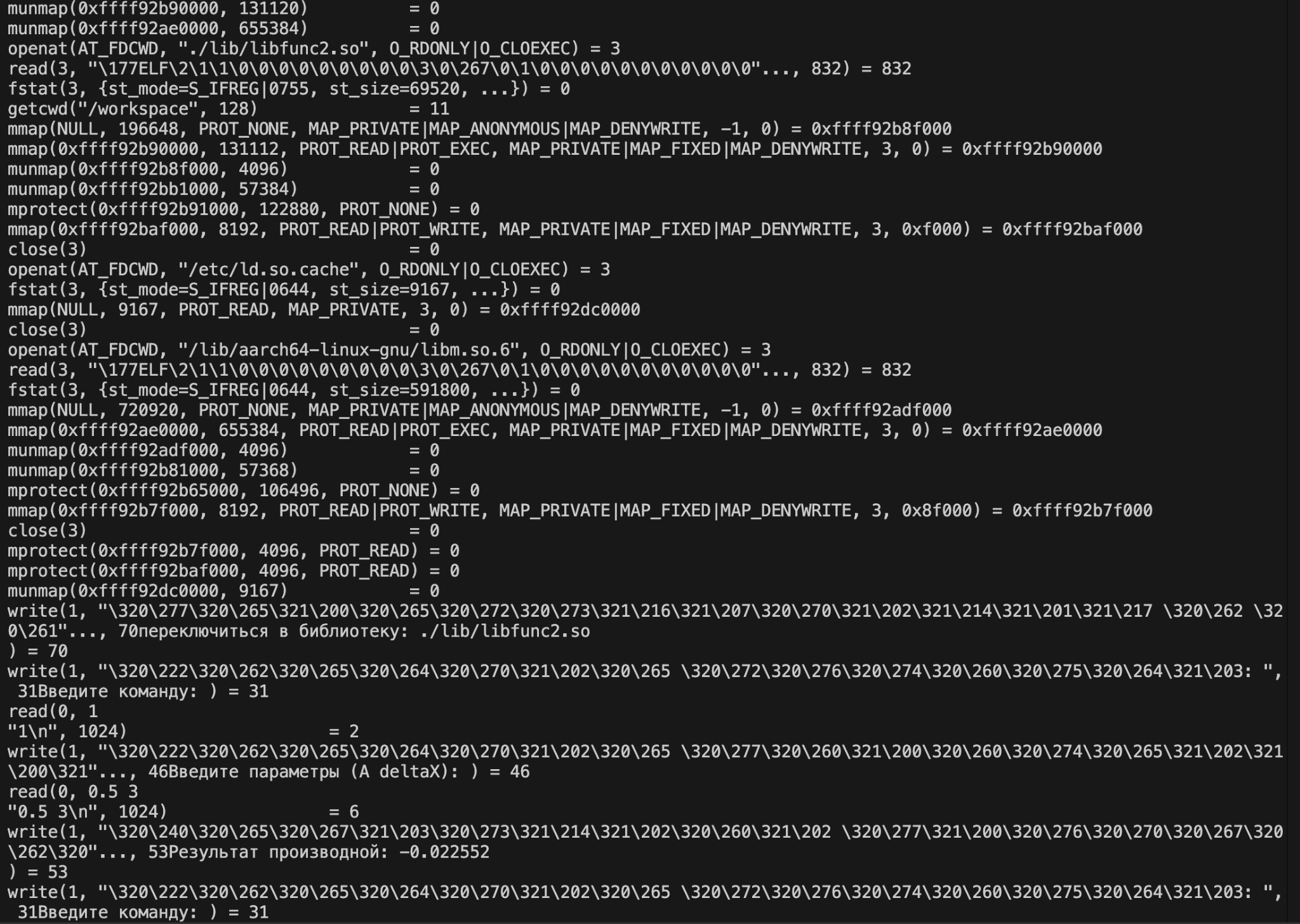
## Лабораторная работа №4.

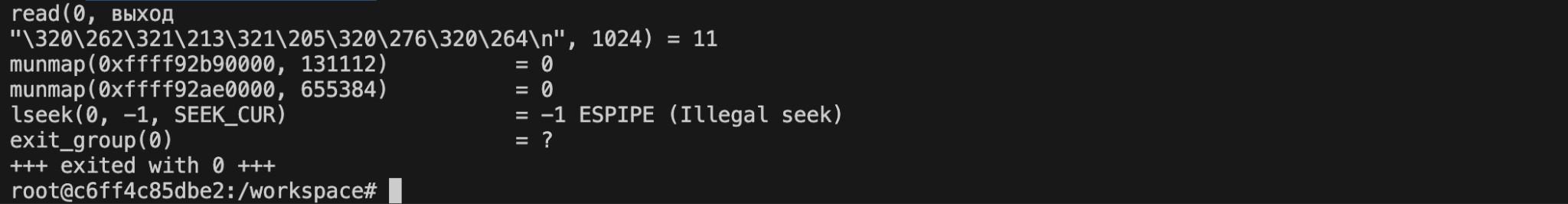
main1.c



main2.c







# Вывод

При выполнении лабораторных работ по курсу “Операционные Системы” мне часто приходилось пользоваться утилитой strace. Она показывает выполняемые процессы, команды, взаимосвязи: одним словом, необходимые разработчику вещи для отладки любой программы. А наличие у strace возможности задавать флаги, указывающие, что именно следует сохранять, делает эту утилиту особенно удобной! Мне было приятно познакомиться с strace и я буду и дальше использовать его в работе.