

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

**Курсовой проект
по курсу «Операционные системы»**

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

Москва, 2025

Условие

Цель курсового проекта:

1. Приобретение практических навыков в использовании знаний, полученных в течении курса
2. Проведение исследования в выбранной предметной области

Задание: Необходимо спроектировать и реализовать программный прототип в соответствии с выбранным вариантом. Произвести анализ и сделать вывод на основании данных, полученных при работе программного прототипа.

Консоль-серверная игра. Необходимо написать консоль-серверную игру. Необходимо написать 2 программы: сервер и клиент. Сначала запускается сервер, а далее клиенты соединяются с сервером. Сервер координирует клиентов между собой. При запуске клиента игрок может выбрать одно из следующих действий (возможно больше, если предусмотрено вариантом):

- Создать игру, введя ее имя
- Присоединиться к одной из существующих игр по имени игры

Морской бой. Общение между сервером и клиентом необходимо организовать при помощи мемогу тар. Каждый игрок должен при запуске ввести свой логин. Для каждого игрока должна вестись статистика игр (сколько побед/поражений). Игрок может посмотреть свою статистику

Вариант: 5

Метод решения

Решение разработано на основе классической архитектуры клиент-сервер, где центральный сервер координирует взаимодействие между двумя клиентами, играющими в морской бой.

Основным механизмом обмена данными между сервером и клиентами служат файловые отображения памяти (mmap-файлы). Этот подход позволяет различным процессам работать с общей областью памяти через обычный файл в системе, что обеспечивает быстрый обмен данными без необходимости сетевых сокетов. Сервер создаёт файл состояния `server_state.mmap`, в котором размещается полная информация обо всех активных играх. Клиенты открывают этот же файл и получают прямой доступ к данным, находящимся в памяти.

Для защиты общих структур данных от одновременного доступа нескольких процессов используются мьютексы (`pthread_mutex_t`) со специальным флагом `PTHREAD_PROCESS_SHARED`, который позволяет мьютексам работать между процессами, а не только между потоками. Более того, мьютексы инициализируются с типом `PTHREAD_MUTEX_RECURSIVE`, что позволяет одному потоку заблокировать мьютекс несколько раз без `deadlock`'а. Это критически важно для надёжности системы при использовании вложенных функций.

Для эффективного уведомления между процессами об изменении состояния используются условные переменные (`pthread_cond_t`), также настроенные на работу между процессами через флаг `PTHREAD_PROCESS_SHARED`. Когда сервер обрабатывает выстрел и обновляет состояние игры, он отправляет сигнал через условную переменную, что пробуждает ожидающего клиента. Это позволяет избежать интенсивного опроса и обеспечивает отзывчивость системы без излишней нагрузки на процессор.

Система рассчитана на параллельное проведение множества независимых игровых сессий. Архитектура позволяет одновременно обрабатывать до шестнадцати отдельных игр, каждая из которых может включать двух и более игроков. Использование отдельного мьютекса `game_mutex` для каждой игры обеспечивает, что блокировка одной игры не влияет на обработку других. Это позволяет серверу эффективно масштабироваться: при добавлении ещё одной игры просто увеличивается счётчик игр в массиве, без необходимости создания новых системных ресурсов (как это было бы необходимо при использовании отдельных `pipe`'ов или сокетов для каждой пары игроков). Таким образом, один процесс сервера может корректно управлять несколькими десятками клиентов, участвующих в различных играх одновременно.

Общий алгоритм работы системы можно описать в виде следующей последовательности шагов:

- Инициализация сервера: удаление старого файла состояния, создание нового файла размером с `ServerState`, отображение его в память, инициализация мьютексов и условных переменных.
- Запуск клиента: ввод логина игрока, инициализация системы статистики, отображение главного меню.
- Создание игры: первый игрок вводит имя игры, сервер создаёт структуру игры в состоянии `GAME_WAITING`, клиент генерирует случайную расстановку кораблей.
- Присоединение игрока: второй игрок выбирает игру по ID, сервер добавляет его в массив игроков, игра переходит в состояние `GAME_SETUP`, клиент генерирует свою расстановку кораблей.
- Начало игры: игра переходит в состояние `GAME_RUNNING`, первый игрок получает ход первым.
- Обработка выстрела: клиент заполняет структуру выстрела с флагом результата -2, ждёт в условной переменной, сервер обрабатывает выстрел в своём цикле, обновляет доску противника, заполняет результат, отправляет сигнал, клиент пробуждается и выводит результат. Ожидающий хода противника клиент также пробуждается и обновляет свою доску в соответствии с результатом хода соперника.
- Управление ходами: после промаха ход переходит к противнику, после попадания или потопления ход остаётся у игрока.
- Завершение игры: когда все корабли противника потоплены, сервер устанавливает статус `GAME_FINISHED`, обновляет статистику обоих игроков через `update_player_stats`, сохраняет данные в файл `players_stats.db`.
- Просмотр статистики: из главного меню игрок может просмотреть свою статистику, включая количество побед, поражений и процент побед.
- Завершение программы: клиент выбирает выход, закрывает файл состояния, сервер завершает обработку по сигналу `SIGINT`, очищает ресурсы.

Описание программы

Модуль `common.h` содержит определения всех основных структур данных и констант, используемых как сервером, так и клиентами. Здесь определены размеры доски (10 на 10), максимальное количество одновременных игр (16), размеры строк для логинов и имён игр, константы для различных состояний. Важной частью этого модуля являются определения состояний клеток доски (`EMPTY` — пусто, `SHIP` — корабль, `HIT` — попадание, `MISS` — промах) и состояний игры (`GAME_WAITING`, `GAME_SETUP`,

GAME_RUNNING, GAME_FINISHED).

В файле `common.h` определены четыре критически важные структуры. Структура `Board` представляет двумерный массив состояний клеток доски размером 10 на 10. Структура `Player` содержит информацию об одном игроке: его логин, статус готовности, его собственную доску с кораблями и доску, отображающую его выстрелы. Структура `GameState` содержит полное состояние одной игры: идентификатор и имя игры, массив двух игроков, счётчик игроков, индекс текущего игрока, статус игры, индекс победителя, флаг актуальности статистики, структура информации о последнем выстреле, мьютекс для защиты состояния и условную переменную для уведомления об изменениях. Структура `PlayerStats` необходима для хранения и записи статистики каждого игрока и содержит поля: логин игрока, переменные количества его побед, поражений и сыгранных игр.

Структура `ServerState`, размещаемая в `memory-mapped` файле, содержит массив до шестнадцати игр, счётчик активных игр, глобальный мьютекс для защиты списка игр и условную переменную для уведомления об изменении списка.

Модуль `os.h` вместе с реализацией `os_linux.c` предоставляет абстрактный слой для работы с операционной системой. Это позволяет при необходимости легко портировать код на другую платформу. Функции `mmap_create` и `mmap_open` отвечают за создание и открытие файловых отображений памяти соответственно. Функция `mmap_create` предварительно удаляет файл если он существует, затем создаёт новый файл нужного размера, отображает его в память и инициализирует нулями. Функция `mmap_write` выполняет копирование данных в отображённую память и вызывает `msync` для синхронизации с файловой системой, обеспечивая видимость изменений для других процессов.

Модуль синхронизации, состоящий из `sync.h`, `sync_init.h` и `sync.c`, реализует обёртки над POSIX примитивами синхронизации. Функция `os_mutex_init` создаёт мьютекс с необходимыми атрибутами для работы между процессами, включая установку флагов `PTHREAD_PROCESS_SHARED` и `PTHREAD_MUTEX_RECURSIVE`. Функция `os_condvar_timedwait` ожидает сигнала условной переменной с таймаутом, возвращая различные коды результата: `OS_WAIT_SUCCESS` при успехе, `OS_WAIT_TIMEOUT` при истечении времени или `OS_WAIT_ERROR` при ошибке. Это критически важно для предотвращения бесконечного ожидания в случае сбоя сервера.

Модуль игровой логики в файлах `game_logic.h` и `game_logic.c` содержит чистую логику игры, независимую от особенностей синхронизации. Функция `init_board` инициализирует пустую доску. Функция `is_valid_placement` проверяет, может ли быть размещён корабль на доске, учитывая не только сам корабль, но и все соседние клетки, которые согласно правилам морского боя должны быть пусты. Функция `randomize_board` генерирует случайную расстановку стандартного флота из десяти кораблей: одного четырёхпалубного, двух трёхпалубных, трёх двухпалубных и четырёх однопалубных кораблей.

Функция `process_shot` обрабатывает один выстрел, получая доску-цель и координаты, возвращая четыре возможных результата: 0 для промаха, 1 для попадания, 2 для потопления корабля и -1 для ошибки. Функция `is_ship_sunk` проверяет, был ли потоплен корабль, к которому принадлежит указанная клетка, проверяя как горизонтальное, так и вертикальное направления. Функция `all_ships_sunk` проверяет окончание игры, возвращая истину если на доске не осталось ни одного неповреждённого корабля.

Серверные функции, определённые в `server.h` и `server.c`, управляет состоянием всех игр на сервере. Функция `server_init` создаёт или открывает файл состояния, инициализируя его нулями и подготавливая структуру для работы. Функция `create_game` создаёт новую

игру, когда первый игрок выбирает создание, инициализируя структуру игры и устанавливая её в состояние ожидания. Функция `join_game` добавляет второго игрока в существующую игру, проверяя различные условия: игра существует, не полная, не началась. Функция `process_server_shot` выполняет основную логику обработки выстрела: вызывает `process_shot` для обновления доски противника, определяет результат и обновляет состояние игры, включая смену хода или завершение игры.

Модуль статистики в файлах `stats.h` и `stats.c` отвечает за отслеживание побед и поражений игроков. Статистика хранится в бинарном файле `players_stats.db`, содержащем массив структур с информацией о каждом игроке. При инициализации система пытается загрузить существующую базу данных, если файл есть, в противном случае создаётся новая пустая база. Функция `update_player_stats` увеличивает счётчики побед или поражений в зависимости от результата игры и сохраняет изменённую статистику обратно в файл. Функция `display_player_stats` выводит статистику конкретного игрока, включая количество побед, поражений, всего игр и процент побед.

Клиентское приложение `battleship_client.c` предоставляет интерфейс для взаимодействия игрока с системой. Клиент управляет главным меню с опциями создания новой игры, присоединения к существующей или просмотра статистики. После присоединения к игре клиент генерирует случайную расстановку кораблей, отображает доски (собственную и доску выстрелов), обрабатывает ввод координат выстрелов и ожидает результата от сервера через условные переменные с таймаутом. Клиент также отвечает за отображение всех игровых сообщений и результатов игры.

Серверное приложение `battleship_server.c` реализует основной цикл обработки игровых событий. Сервер создаёт и управляет файлом состояния, инициализирует синхронизирующие примитивы для всех игр, и работает в основном цикле. На каждой итерации сервер читает состояние всех активных игр, проверяет наличие новых выстрелов, обрабатывает выстрелы через функцию `process_server_shot`, обновляет доски, определяет результаты, и отправляет сигналы клиентам через условные переменные. Сервер также отвечает за обновление статистики игроков при завершении игры и управление корректным завершением при получении сигнала `SIGINT`.

Результаты

Система успешно реализует полнофункциональную игру морской бой с архитектурой клиент-сервер, координируемую через `memory-mapped` файлы. При запуске сервер создаёт файл состояния, инициализирует примитивы синхронизации и переходит в режим ожидания клиентов. Одновременно играть друг с другом могут до 32 игроков включительно (16 игровых сессий параллельно).

Игровой процесс протекает в соответствии с классическими правилами морского боя. Игроки поочередно делают выстрелы, указывая координаты на доске противника. Каждый выстрел обрабатывается сервером за несколько миллисекунд: проверяется координата, определяется результат (промах/попадание/потопление), обновляется состояние досок обоих игроков. Система правильно выводит результаты выстрелов, управляет чередованием ходов согласно правилам (ход остаётся при попадании, переходит при промахе), и отображает обновленные доски для каждого игрока. При потоплении корабля система корректно определяет это по окружающим клеткам, а при потоплении всех кораблей объявляет победителя.

По завершении игры сервер обновляет статистику обоих игроков, сохраняя количество побед и поражений в файл `players_stats.db`. Игроки могут вернуться в главное меню и

просмотреть свою статистику с указанием общего количества побед, поражений, всего сыгранных игр и процента побед. Данные статистики сохраняются между запусками программы, что позволяет игрокам отслеживать свой прогресс. Система также обеспечивает контролируемое завершение: клиент может выбрать выход из программы, а сервер корректно завершает работу при получении сигнала SIGINT (Ctrl+C), очищая все ресурсы и синхронизирующие примитивы.

Выводы

Цели курсового проекта достигнуты, задача, поставленная передо мной успешно выполнена. Знания, полученные при изучении курса "Операционные системы" применены на практике для создания цельного многопользовательского программного прототипа готового к надежному использованию.

Курсовая работа полностью решает поставленную задачу разработки консоль-серверной игры "Морской бой" с использованием технологии memogu map в соответствии с вариантом 5. Реализована архитектура клиент-сервер, где сервер координирует взаимодействие двух клиентов через память, отображённую в файл, обеспечивая быстрый обмен данными и надёжную синхронизацию. Система демонстрирует глубокое понимание функционирования и взаимодействия процессов и механизмов синхронизации между ними.

Логика игры реализована полностью и корректно, включая проверку валидности размещения кораблей с учётом соседних клеток, определение потопления по обоим направлениям и окончания игры. Система статистики надёжно отслеживает побед и поражений каждого игрока с сохранением данных между запусками в бинарном файле.

С точки зрения производительности система демонстрирует хорошие характеристики для своего предназначения. Цикл сервера работает с периодом 500 миллисекунд, что обеспечивает достаточную отзывчивость при обработке выстрелов. При типичном времени игры в несколько минут эта задержка незначительна и не влияет на качество игрового опыта. Таймаут ожидания клиента составляет 300 секунд, что достаточно для любого человека, чтобы сделать ход, и в то же время позволяет обнаружить сбой сервера.

Использование memogu-mapped файлов обеспечивает минимальные накладные расходы на копирование данных. Вместо пересылки данных через pipe или через очередь сообщений, процессы работают с одной и той же областью памяти. Это критически важно для систем с требованиями к низким задержкам и экономии ресурсов.

Память, используемая системой, достаточно предсказуема и не растёт с временем выполнения. Сервер выделяет один файл состояния, размер которого определяется максимальным количеством игр и размером структуры игры. Каждый клиент выделяет в памяти копию состояния для быстрого доступа, но размер этой копии фиксирован и не зависит от количества игроков или времени игры.

Система демонстрирует хорошую надёжность благодаря нескольким механизмам обработки ошибок. Инициализация синхронизирующих примитивов проверяется на каждом этапе, что предотвращает ситуации, когда система работает с неинициализированными примитивами синхронизации.

Клиент использует таймаут при ожидании результата выстрела, что предотвращает ситуацию, когда клиент зависает бесконечно и не отзывается на действия пользователя.

Валидация координат выстрела проверяется как на клиенте (перед отправкой), так и на сервере (при обработке). Тем самым исключается возможность неверной обработки или подмены данных со стороны клиента.

При контролируемом завершении сервера (нажатие Ctrl+C) все активные игры корректно завершаются, статус устанавливается в завершённую, синхронизирующие примитивы очищаются, и файл закрывается.

Разработка этого проекта закрепила и расширила множество практических навыков в области системного программирования:

- Память и файловые операции. Практическое использование функций "mmap", "munmap" и "msync" укрепили знания о том, как операционная система управляет отображением файлов в память и синхронизацией с диском.
- POSIX синхронизация. Глубокое овладение мьютексами и условными переменными. Понимание критических ошибок синхронизации (race conditions, deadlocks) стало интуитивным через практическое выявление и исправление таких ошибок.
- Межпроцессное взаимодействие. Освоены различные подходы к IPC и их компромиссы между производительностью и сложностью реализации.
- Архитектурное проектирование. Навыки проектирования многопроцессных систем, выбор между централизованной и распределённой синхронизацией, управление состоянием множества независимых сущностей. Понимание компромиссов между простотой и производительностью при проектировании архитектуры.
- Многопроцессная отладка. Использование инструментов отладки ("strace", логирование состояния) для понимания взаимодействия процессов на низком уровне.

Исходная программа

```
1  #ifndef COMMON_H
2  #define COMMON_H
3
4  #include <stdio.h>
5  #include <string.h>
6
7  #include "os.h"
8  #include "sync.h"
9
10 #define BOARD_SIZE 10
11 #define MAX_GAMES 16
12 #define MAX_PLAYERS_PER_GAME 2
13 #define MAX_GAME_NAME 128
14 #define MAX_LOGIN 64
15 #define MAX_GAME_ID 32
16
17 typedef enum { EMPTY = 0, SHIP = 1, HIT = 2, MISS = 3 } CellState;
18
19 typedef enum {
20     GAME_WAITING = 0,
21     GAME_SETUP = 1,
22     GAME_RUNNING = 2,
23     GAME_FINISHED = 3
24 } GameStatus;
25
26 typedef struct {
27     CellState cells[BOARD_SIZE][BOARD_SIZE];
28 } Board;
29
30 typedef struct {
31     char login[MAX_LOGIN];
32     int ready;
33     Board board;
34     Board shots;
35 } Player;
36
37 typedef struct {
38     char game_id[MAX_GAME_ID];
39     char game_name[MAX_GAME_NAME];
40     Player players[MAX_PLAYERS_PER_GAME];
41     int player_count;
42     int current_turn;
43     int status;
44     int winner_idx;
45     int stats_updated;
46
47     struct {
48         int active;
49         int row;
50         int col;
```

```

51     int result;
52     int processed_by_opponent;
53 } last_shots[MAX_PLAYERS_PER_GAME];
54
55     int last_update_version;
56
57     OSSyncMutex game_mutex;
58     OSSyncCondVar shot_changed;
59 } GameState;
60
61 typedef struct {
62     GameState games[MAX_GAMES];
63     int game_count;
64
65     OSSyncMutex state_mutex;
66     OSSyncCondVar state_changed;
67 } ServerState;
68
69 typedef struct {
70     char login[MAX_LOGIN];
71     int wins;
72     int losses;
73     int games_played;
74 } PlayerStats;
75
76 #endif // COMMON_H

```

Листинг 1: *Заголовочный файл определяющий основные структуры данных и константы*

```

1  #ifndef OS_H
2  #define OS_H
3
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <string.h>
7  #include <time.h>
8
9  #ifdef _WIN32
10 #define OS_WINDOWS 1
11 #define OS_LINUX 0
12 #else
13 #define OS_WINDOWS 0
14 #define OS_LINUX 1
15 #endif
16
17 typedef struct {
18     void *addr;
19     int fd;
20 } OSMmapHandle;
21

```

```

22 | OSMmapHandle mmap_create(const char *filename, size_t size);
23 |
24 | OSMmapHandle mmap_open(const char *filename, size_t size);
25 |
26 | void mmap_close(OSMmapHandle handle, size_t size);
27 |
28 | void mmap_write(void *addr, const void *data, size_t size);
29 |
30 | void mmap_read(void *addr, void *data, size_t size);
31 |
32 | void os_file_unlink(const char *filename);
33 |
34 | void init_random(void);
35 |
36 | int os_get_pid(void);
37 |
38 | void os_usleep(unsigned int microseconds);
39 |
40 | typedef void (*OSSignalHandler)(int);
41 |
42 | void os_signal_register_int(OSSignalHandler handler);
43 |
44 | #endif // OS_H

```

Листинг 2: Объявление абстрактного слоя для кроссплатформенности (системные вызовы и работа с файлами)

```

1 | #ifndef SYNC_H
2 | #define SYNC_H
3 |
4 | #include <pthread.h>
5 |
6 | #ifdef _WIN32
7 | #else
8 | typedef pthread_mutex_t OSSyncMutex;
9 | typedef pthread_cond_t OSSyncCondVar;
10 | #endif
11 |
12 | int os_mutex_init(OSSyncMutex *mutex);
13 |
14 | void os_mutex_destroy(OSSyncMutex *mutex);
15 |
16 | void os_mutex_lock(OSSyncMutex *mutex);
17 |
18 | void os_mutex_unlock(OSSyncMutex *mutex);
19 |
20 | int os_condvar_init(OSSyncCondVar *cond);
21 |
22 | void os_condvar_destroy(OSSyncCondVar *cond);
23 |

```

```

24 | int os_condvar_timedwait(OSSyncCondVar *cond, OSSyncMutex *mutex,
25 |                         int timeout_ms);
26 |
27 | void os_condvar_broadcast(OSSyncCondVar *cond);
28 |
29 | void os_condvar_signal(OSSyncCondVar *cond);
30 |
31 | #define OS_WAIT_SUCCESS 0
32 | #define OS_WAIT_TIMEOUT 1
33 | #define OS_WAIT_ERROR -1
34 |
35 | #endif // SYNC_H

```

Листинг 3: Объявление абстрактного слоя для кроссплатформенности (примитивы синхронизации)

```

1 | #ifndef MUTEX_INIT_H
2 | #define MUTEX_INIT_H
3 |
4 | #include "common.h"
5 |
6 | int init_server_mutexes(ServerState *state);
7 |
8 | int init_game_mutexes(GameState *game);
9 |
10 | void cleanup_server_mutexes(ServerState *state);
11 |
12 | #endif

```

Листинг 4: *Объявление функций инициализации примитивов синхронизации*

```

1 | #ifndef STATS_H
2 | #define STATS_H
3 |
4 | #include "common.h"
5 |
6 | #define STATS_FILE "players_stats.db"
7 | #define MAX_PLAYERS_STATS 1000
8 |
9 | int stats_init();
10 |
11 | int load_player_stats(const char *login, PlayerStats *stats);
12 |
13 | int save_player_stats(PlayerStats *stats);
14 |
15 | int update_player_stats(const char *login, int is_win, int is_loss);
16 |
17 | void display_player_stats(const char *login);
18 |

```

```
19 | #endif // STATS_H
```

Листинг 5: *Заголовочный файл отслеживания статистики*

```
1 | #ifndef SERVER_H
2 | #define SERVER_H
3 |
4 | #include "common.h"
5 |
6 | #define STATE_FILE "server_state.mmap"
7 |
8 | int server_init(ServerState **state, OSMmapHandle state_handle);
9 |
10 | void server_cleanup(ServerState *state, OSMmapHandle state_handle);
11 |
12 | int create_game(ServerState *state, const char *game_name,
13 |               const char *player_login, char *out_game_id);
14 |
15 | int join_game(ServerState *state, const char *game_id,
16 |              const char *player_login);
17 |
18 | GameState *get_game(ServerState *state, const char *game_id);
19 |
20 | void list_games(ServerState *state);
21 |
22 | int check_game_ready(GameState *game);
23 |
24 | void finish_game(GameState *game, int winner_idx);
25 |
26 | int process_server_shot(GameState *game, int shooter_idx, int row, int col);
27 |
28 | #endif // SERVER_H
```

Листинг 6: *Заголовочный файл серверных функций*

```
1 | #ifndef GAME_LOGIC_H
2 | #define GAME_LOGIC_H
3 |
4 | #include "common.h"
5 |
6 | #define MAX_ATTEMPTS 100
7 |
8 | void place_ship(Board *board, int row, int col, int length, int horizontal);
9 |
10 | int is_valid_placement(Board *board, int row, int col, int length,
11 |                       int horizontal);
12 |
13 | void randomize_board(Board *board);
14 |
```

```

15 void init_board(Board *board);
16
17 int process_shot(Board *target_board, int row, int col);
18
19 int is_ship_sunk(Board *board, int row, int col);
20
21 int all_ships_sunk(Board *board);
22
23 char cell_to_char(CellState state, int reveal_ships);
24
25 #endif // GAME_LOGIC_H

```

Листинг 7: *Заголовочный файл игровой логики*

```

1  #include "os.h"
2
3  #if OS_LINUX
4
5  #include <fcntl.h>
6  #include <signal.h>
7  #include <sys/mman.h>
8  #include <sys/stat.h>
9  #include <time.h>
10
11 OSMmapHandle mmap_create(const char *filename, size_t size) {
12     OSMmapHandle handle = {NULL, -1};
13     unlink(filename);
14
15     int fd = open(filename, O_CREAT | O_RDWR | O_TRUNC, 0666);
16     if (fd < 0) {
17         perror("open");
18         return handle;
19     }
20
21     if (lseek(fd, size - 1, SEEK_SET) == -1) {
22         perror("lseek");
23         close(fd);
24         return handle;
25     }
26
27     if (write(fd, "", 1) != 1) {
28         perror("write");
29         close(fd);
30         return handle;
31     }
32
33     void *addr = mmap(NULL, size, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
34     if (addr == MAP_FAILED) {
35         perror("mmap");
36         close(fd);

```

```

37     return handle;
38 }
39
40 memset(addr, 0, size);
41 handle.addr = addr;
42 handle.fd = fd;
43
44 return handle;
45 }
46
47 OSMmapHandle mmap_open(const char *filename, size_t size) {
48     OSMmapHandle handle = {NULL, -1};
49
50     int fd = open(filename, O_RDWR, 0666);
51     if (fd < 0) {
52         perror("open");
53         return handle;
54     }
55
56     void *addr = mmap(NULL, size, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
57     if (addr == MAP_FAILED) {
58         perror("mmap");
59         close(fd);
60         return handle;
61     }
62
63     handle.addr = addr;
64     handle.fd = fd;
65
66     return handle;
67 }
68
69 void mmap_close(OSMmapHandle handle, size_t size) {
70     if (handle.addr != NULL && handle.addr != MAP_FAILED) {
71         munmap(handle.addr, size);
72     }
73
74     if (handle.fd >= 0) {
75         close(handle.fd);
76     }
77 }
78
79 void mmap_write(void *addr, const void *data, size_t size) {
80     if (addr && data) {
81         memcpy(addr, data, size);
82         msync(addr, size, MS_SYNC);
83     }
84 }
85
86 void mmap_read(void *addr, void *data, size_t size) {
87     if (addr && data) {

```

```

88     memcpy(data, addr, size);
89 }
90 }
91
92 void os_file_unlink(const char *filename) { unlink(filename); }
93
94 void init_random(void) { srand((unsigned int)time(NULL) ^ getpid()); }
95
96 int os_get_pid(void) { return getpid(); }
97
98 void os_usleep(unsigned int microseconds) { usleep(microseconds); }
99
100 void os_signal_register_int(OSSignalHandler handler) {
101     signal(SIGINT, handler);
102 }
103
104 #endif // OS_LINUX

```

Листинг 8: Реализация абстрактного слоя для кроссплатформенности (POSIX системные вызовы и работа с файлами)

```

1  #include "sync.h"
2
3  #include <stdio.h>
4
5  #include "sync_init.h"
6
7  #ifdef _WIN32
8  #else
9  #include <errno.h>
10 #include <pthread.h>
11 #include <time.h>
12 #endif
13
14 int os_mutex_init(OSSyncMutex *mutex) {
15     pthread_mutexattr_t attr;
16
17     if (pthread_mutexattr_init(&attr) != 0) {
18         perror("pthread_mutexattr_init");
19         return -1;
20     }
21
22     if (pthread_mutexattr_setpshared(&attr, PTHREAD_PROCESS_SHARED) != 0) {
23         perror("pthread_mutexattr_setpshared");
24         pthread_mutexattr_destroy(&attr);
25         return -1;
26     }
27
28     if (pthread_mutexattr_settype(&attr, PTHREAD_MUTEX_RECURSIVE) != 0) {
29         perror("pthread_mutexattr_settype");

```

```

30     pthread_mutexattr_destroy(&attr);
31     return -1;
32 }
33
34 int result = pthread_mutex_init(mutex, &attr);
35 pthread_mutexattr_destroy(&attr);
36 return result;
37 }
38
39 void os_mutex_destroy(OSSyncMutex *mutex) {
40     if (mutex) {
41         pthread_mutex_destroy(mutex);
42     }
43 }
44
45 void os_mutex_lock(OSSyncMutex *mutex) {
46     if (mutex) {
47         pthread_mutex_lock(mutex);
48     }
49 }
50
51 void os_mutex_unlock(OSSyncMutex *mutex) {
52     if (mutex) {
53         pthread_mutex_unlock(mutex);
54     }
55 }
56
57 int os_condvar_init(OSSyncCondVar *cond) {
58     pthread_condattr_t attr;
59
60     if (pthread_condattr_init(&attr) != 0) {
61         perror("pthread_condattr_init");
62         return -1;
63     }
64
65     if (pthread_condattr_setpshared(&attr, PTHREAD_PROCESS_SHARED) != 0) {
66         perror("pthread_condattr_setpshared");
67         pthread_condattr_destroy(&attr);
68         return -1;
69     }
70
71     int result = pthread_cond_init(cond, &attr);
72     pthread_condattr_destroy(&attr);
73     return result;
74 }
75
76 void os_condvar_destroy(OSSyncCondVar *cond) {
77     if (cond) {
78         pthread_cond_destroy(cond);
79     }
80 }

```

```

81
82 int os_condvar_timedwait(OSSyncCondVar *cond, OSSyncMutex *mutex,
83                          int timeout_ms) {
84     if (!cond || !mutex) {
85         return OS_WAIT_ERROR;
86     }
87
88     struct timespec timeout;
89     if (clock_gettime(CLOCK_REALTIME, &timeout) != 0) {
90         perror("clock_gettime");
91         return OS_WAIT_ERROR;
92     }
93
94     long ns = (long)timeout.tv_nsec + ((long)timeout_ms * 1000000);
95     timeout.tv_sec += ns / 1000000000;
96     timeout.tv_nsec = ns % 1000000000;
97
98     int result = pthread_cond_timedwait(cond, mutex, &timeout);
99
100    if (result == ETIMEDOUT) {
101        return OS_WAIT_TIMEOUT;
102    } else if (result == 0) {
103        return OS_WAIT_SUCCESS;
104    } else {
105        return OS_WAIT_ERROR;
106    }
107 }
108
109 void os_condvar_broadcast(OSSyncCondVar *cond) {
110     if (cond) {
111         pthread_cond_broadcast(cond);
112     }
113 }
114
115 void os_condvar_signal(OSSyncCondVar *cond) {
116     if (cond) {
117         pthread_cond_signal(cond);
118     }
119 }
120
121 int init_server_mutexes(ServerState *state) {
122     if (!state) return 0;
123
124     printf("Initialized state_mutex\n");
125     if (os_mutex_init(&state->state_mutex) != 0) {
126         return 0;
127     }
128
129     printf("Initialized state_changed condition variable\n");
130     if (os_condvar_init(&state->state_changed) != 0) {
131         os_mutex_destroy(&state->state_mutex);

```

```

132     return 0;
133 }
134
135 return 1;
136 }
137
138 int init_game_mutexes(GameState *game) {
139     if (!game) return 0;
140
141     if (os_mutex_init(&game->game_mutex) != 0) {
142         return 0;
143     }
144
145     if (os_condvar_init(&game->shot_changed) != 0) {
146         os_mutex_destroy(&game->game_mutex);
147         return 0;
148     }
149
150     return 1;
151 }
152
153 void cleanup_server_mutexes(ServerState *state) {
154     if (!state) return;
155
156     for (int i = 0; i < state->game_count; i++) {
157         os_mutex_destroy(&state->games[i].game_mutex);
158         os_condvar_destroy(&state->games[i].shot_changed);
159     }
160
161     os_mutex_destroy(&state->state_mutex);
162     os_condvar_destroy(&state->state_changed);
163
164     printf("Synchronization primitives cleaned up\n");
165 }

```

Листинг 9: *Реализация абстрактного слоя примитивов синхронизации (POSIX)*

```

1  #include "stats.h"
2
3  typedef struct {
4      PlayerStats players[MAX_PLAYERS_STATS];
5      int count;
6  } StatsDatabase;
7
8  static StatsDatabase db;
9  static int db_initialized = 0;
10
11 int stats_init() {
12     memset(&db, 0, sizeof(StatsDatabase));
13

```

```

14 FILE *f = fopen(STATS_FILE, "rb");
15 if (f) {
16     fread(&db, sizeof(StatsDatabase), 1, f);
17     fclose(f);
18 }
19
20 db_initialized = 1;
21 return 1;
22 }
23
24 static int find_or_create_player(const char *login) {
25     for (int i = 0; i < db.count; i++) {
26         if (strcmp(db.players[i].login, login) == 0) {
27             return i;
28         }
29     }
30
31     if (db.count >= MAX_PLAYERS_STATS) {
32         fprintf(stderr, "Error: statistics database is full\n");
33         return -1;
34     }
35
36     int idx = db.count;
37     memset(&db.players[idx], 0, sizeof(PlayerStats));
38     strcpy(db.players[idx].login, login);
39     db.count++;
40     return idx;
41 }
42
43 int load_player_stats(const char *login, PlayerStats *stats) {
44     if (!db_initialized) stats_init();
45
46     int idx = find_or_create_player(login);
47     if (idx < 0) return 0;
48
49     *stats = db.players[idx];
50     return 1;
51 }
52
53 int save_player_stats(PlayerStats *stats) {
54     if (!db_initialized) stats_init();
55
56     int idx = find_or_create_player(stats->login);
57     if (idx < 0) return 0;
58
59     db.players[idx] = *stats;
60
61     FILE *f = fopen(STATS_FILE, "wb");
62     if (!f) {
63         perror("fopen");
64         return 0;

```

```

65     }
66
67     fwrite(&db, sizeof(StatsDatabase), 1, f);
68     fclose(f);
69     return 1;
70 }
71
72 int update_player_stats(const char *login, int is_win, int is_loss) {
73     if (!db_initialized) stats_init();
74
75     PlayerStats stats;
76     load_player_stats(login, &stats);
77
78     if (is_win) {
79         stats.wins++;
80         stats.games_played++;
81     } else if (is_loss) {
82         stats.losses++;
83         stats.games_played++;
84     }
85
86     return save_player_stats(&stats);
87 }
88
89 void display_player_stats(const char *login) {
90     if (!db_initialized) stats_init();
91
92     PlayerStats stats;
93     load_player_stats(login, &stats);
94
95     printf("\n===== PLAYER STATISTICS =====\n");
96     printf("Login: %s\n", stats.login);
97     printf("Wins: %d\n", stats.wins);
98     printf("Losses: %d\n", stats.losses);
99     printf("Total games: %d\n", stats.games_played);
100
101     if (stats.games_played > 0) {
102         double win_rate = (double)stats.wins / stats.games_played * 100;
103         printf("Win rate: %.1f%%\n", win_rate);
104     }
105 }

```

Листинг 10: *Реализация отслеживания статистики*

```

1  #include "server.h"
2
3  #include "game_logic.h"
4  #include "stats.h"
5  #include "sync_init.h"
6

```

```

7  int server_init(ServerState **state, OSMmapHandle state_handle) {
8      os_file_unlink(STATE_FILE);
9      state_handle = mmap_create(STATE_FILE, sizeof(ServerState));
10     *state = (ServerState *)state_handle.addr;
11
12     if (*state == NULL) {
13         fprintf(stderr, "Error: failed to create state file\n");
14         return 0;
15     }
16
17     (*state)->game_count = 0;
18     printf("Server initialized\n");
19     return 1;
20 }
21
22 void server_cleanup(ServerState *state, OSMmapHandle handle) {
23     mmap_close(handle, sizeof(ServerState));
24 }
25
26 int create_game(ServerState *state, const char *game_name,
27                 const char *player_login, char *out_game_id) {
28     os_mutex_lock(&state->state_mutex);
29
30     if (state->game_count >= MAX_GAMES) {
31         fprintf(stderr, "Error: maximum number of games reached\n");
32         os_mutex_unlock(&state->state_mutex);
33         return 0;
34     }
35
36     GameState *game = &state->games[state->game_count];
37     snprintf(out_game_id, MAX_GAME_ID, "%d", state->game_count);
38     strcpy(game->game_id, out_game_id);
39     strcpy(game->game_name, game_name);
40
41     strcpy(game->players[0].login, player_login);
42     game->players[0].ready = 0;
43     init_board(&game->players[0].board);
44     init_board(&game->players[0].shots);
45
46     game->player_count = 1;
47     game->current_turn = 0;
48     game->status = GAME_WAITING;
49     game->winner_idx = -1;
50     game->stats_updated = 0;
51
52     for (int i = 0; i < 2; i++) {
53         game->last_shots[i].result = -1;
54         game->last_shots[i].processed_by_opponent = 0;
55     }
56
57     if (!init_game_mutexes(game)) {

```

```

58     fprintf(stderr, "Error initializing game mutexes\n");
59     os_mutex_unlock(&state->state_mutex);
60     return 0;
61 }
62
63 game->last_update_version = 0;
64 state->game_count++;
65 os_condvar_broadcast(&state->state_changed);
66 os_mutex_unlock(&state->state_mutex);
67
68 printf("Game '%s' created (ID: %s)\n", game_name, out_game_id);
69 return 1;
70 }
71
72 int join_game(ServerState *state, const char *game_id,
73              const char *player_login) {
74     os_mutex_lock(&state->state_mutex);
75
76     for (int i = 0; i < state->game_count; i++) {
77         if (strcmp(state->games[i].game_id, game_id) == 0) {
78             GameState *game = &state->games[i];
79
80             if (game->player_count >= MAX_PLAYERS_PER_GAME) {
81                 fprintf(stderr, "Error: game is full\n");
82                 os_mutex_unlock(&state->state_mutex);
83                 return 0;
84             }
85
86             if (game->status != GAME_WAITING) {
87                 fprintf(stderr, "Error: game already started\n");
88                 os_mutex_unlock(&state->state_mutex);
89                 return 0;
90             }
91
92             os_mutex_lock(&game->game_mutex);
93
94             if (game->player_count >= MAX_PLAYERS_PER_GAME) {
95                 fprintf(stderr, "Error: game is full\n");
96                 os_mutex_unlock(&game->game_mutex);
97                 os_mutex_unlock(&state->state_mutex);
98                 return 0;
99             }
100
101             strcpy(game->players[1].login, player_login);
102             game->players[1].ready = 0;
103             init_board(&game->players[1].board);
104             init_board(&game->players[1].shots);
105             game->player_count++;
106             game->status = GAME_SETUP;
107
108             os_condvar_broadcast(&game->shot_changed);

```

```

109     os_condvar_broadcast(&state->state_changed);
110     os_mutex_unlock(&game->game_mutex);
111     os_mutex_unlock(&state->state_mutex);
112
113     printf("Player '%s' joined the game\n", player_login);
114     return 1;
115 }
116 }
117
118 os_mutex_unlock(&state->state_mutex);
119 fprintf(stderr, "Error: game not found\n");
120 return 0;
121 }
122
123 GameState *get_game(ServerState *state, const char *game_id) {
124     for (int i = 0; i < state->game_count; i++) {
125         if (strcmp(state->games[i].game_id, game_id) == 0) {
126             return &state->games[i];
127         }
128     }
129     return NULL;
130 }
131
132 void list_games(ServerState *state) {
133     printf("\n===== AVAILABLE GAMES =====\n");
134
135     if (state->game_count == 0) {
136         printf("No available games\n");
137         return;
138     }
139
140     for (int i = 0; i < state->game_count; i++) {
141         GameState *game = &state->games[i];
142
143         if (game->status == GAME_FINISHED) {
144             continue;
145         }
146
147         printf("%d. [%s] %s - %d/%d players (status: ", i + 1, game->game_id,
148             game->game_name, game->player_count, MAX_PLAYERS_PER_GAME);
149
150         switch (game->status) {
151             case GAME_WAITING:
152                 printf("waiting");
153                 break;
154             case GAME_SETUP:
155                 printf("setup");
156                 break;
157             case GAME_RUNNING:
158                 printf("running");
159                 break;

```

```

160         case GAME_FINISHED:
161             printf("finished");
162             break;
163         default:
164             printf("?");
165             break;
166     }
167     printf("\n");
168 }
169 }
170
171 int check_game_ready(GameState *game) {
172     if (game->player_count != 2) {
173         return 0;
174     }
175
176     if (game->players[0].ready && game->players[1].ready) {
177         game->status = GAME_RUNNING;
178         return 1;
179     }
180
181     return 0;
182 }
183
184 void finish_game(GameState *game, int winner_idx) {
185     if (game->status != GAME_RUNNING) {
186         return;
187     }
188
189     game->status = GAME_FINISHED;
190     game->winner_idx = winner_idx;
191
192     if (winner_idx >= 0 && winner_idx < 2) {
193         update_player_stats(game->players[winner_idx].login, 1, 0);
194         int loser_idx = 1 - winner_idx;
195         update_player_stats(game->players[loser_idx].login, 0, 1);
196         printf("Game finished! Winner: %s\n", game->players[winner_idx].login);
197     }
198 }
199
200 int process_server_shot(GameState *game, int shooter_idx, int row, int col) {
201     if (game->status != GAME_RUNNING) {
202         return -1;
203     }
204
205     int opponent_idx = 1 - shooter_idx;
206
207     if (row < 0 || row >= BOARD_SIZE || col < 0 || col >= BOARD_SIZE) {
208         return -1;
209     }
210

```

```

211 Board *shooter_shots = &game->players[shooter_idx].shots;
212
213 if (shooter_shots->cells[row][col] != EMPTY) {
214     return -1;
215 }
216
217 Board *opponent_board = &game->players[opponent_idx].board;
218 int result = process_shot(opponent_board, row, col);
219
220 if (result == -1) {
221     return -1;
222 }
223
224 if (result == 0) {
225     shooter_shots->cells[row][col] = MISS;
226 } else {
227     shooter_shots->cells[row][col] = HIT;
228 }
229
230 game->players[opponent_idx].board = *opponent_board;
231 game->players[shooter_idx].shots = *shooter_shots;
232
233 printf("[SHOT] Player %d shoots [%d,%d] -> ", shooter_idx, row, col);
234
235 if (result == 0) {
236     game->current_turn = opponent_idx;
237     printf("MISS. Turn goes to player %d\n", opponent_idx);
238 } else if (result == 1) {
239     game->current_turn = shooter_idx;
240     printf("HIT! Turn stays with player %d\n", shooter_idx);
241 } else if (result == 2) {
242     game->current_turn = shooter_idx;
243     printf("SUNK! Turn stays with player %d\n", shooter_idx);
244 }
245
246 if (all_ships_sunk(opponent_board)) {
247     game->status = GAME_FINISHED;
248     game->winner_idx = shooter_idx;
249     printf("[VICTORY] Player %d won!\n", shooter_idx);
250     return 3;
251 }
252
253 return result;
254 }

```

Листинг 11: *Реализация серверных функций*

```

1 #include "game_logic.h"
2
3 void init_board(Board *board) {

```

```

4   for (int i = 0; i < BOARD_SIZE; i++) {
5       for (int j = 0; j < BOARD_SIZE; j++) {
6           board->cells[i][j] = EMPTY;
7       }
8   }
9 }
10
11 int is_valid_placement(Board *board, int row, int col, int length,
12                        int horizontal) {
13     if (horizontal) {
14         if (col + length > BOARD_SIZE) return 0;
15     } else {
16         if (row + length > BOARD_SIZE) return 0;
17     }
18
19     for (int i = 0; i < length; i++) {
20         int r = horizontal ? row : row + i;
21         int c = horizontal ? col + i : col;
22
23         if (board->cells[r][c] != EMPTY) {
24             return 0;
25         }
26     }
27
28     if (horizontal) {
29         if (col > 0) {
30             for (int r = row - 1; r <= row + 1; r++) {
31                 if (r >= 0 && r < BOARD_SIZE) {
32                     if (board->cells[r][col - 1] != EMPTY) return 0;
33                 }
34             }
35         }
36
37         if (col + length < BOARD_SIZE) {
38             for (int r = row - 1; r <= row + 1; r++) {
39                 if (r >= 0 && r < BOARD_SIZE) {
40                     if (board->cells[r][col + length] != EMPTY) return 0;
41                 }
42             }
43         }
44
45         if (row > 0) {
46             for (int c = col - 1; c <= col + length; c++) {
47                 if (c >= 0 && c < BOARD_SIZE) {
48                     if (board->cells[row - 1][c] != EMPTY) return 0;
49                 }
50             }
51         }
52         if (row < BOARD_SIZE - 1) {
53             for (int c = col - 1; c <= col + length; c++) {
54                 if (c >= 0 && c < BOARD_SIZE) {

```

```

55         if (board->cells[row + 1][c] != EMPTY) return 0;
56     }
57 }
58 }
59 } else {
60     if (row > 0) {
61         for (int c = col - 1; c <= col + 1; c++) {
62             if (c >= 0 && c < BOARD_SIZE) {
63                 if (board->cells[row - 1][c] != EMPTY) return 0;
64             }
65         }
66     }
67
68     if (row + length < BOARD_SIZE) {
69         for (int c = col - 1; c <= col + 1; c++) {
70             if (c >= 0 && c < BOARD_SIZE) {
71                 if (board->cells[row + length][c] != EMPTY) return 0;
72             }
73         }
74     }
75
76     if (col > 0) {
77         for (int r = row - 1; r <= row + length; r++) {
78             if (r >= 0 && r < BOARD_SIZE) {
79                 if (board->cells[r][col - 1] != EMPTY) return 0;
80             }
81         }
82     }
83     if (col < BOARD_SIZE - 1) {
84         for (int r = row - 1; r <= row + length; r++) {
85             if (r >= 0 && r < BOARD_SIZE) {
86                 if (board->cells[r][col + 1] != EMPTY) return 0;
87             }
88         }
89     }
90 }
91
92 return 1;
93 }
94
95 void place_ship(Board *board, int row, int col, int length, int horizontal) {
96     if (!is_valid_placement(board, row, col, length, horizontal)) {
97         return;
98     }
99
100     for (int i = 0; i < length; i++) {
101         int r = horizontal ? row : row + i;
102         int c = horizontal ? col + i : col;
103         board->cells[r][c] = SHIP;
104     }
105 }

```

```

106
107 void randomize_board(Board *board) {
108     init_board(board);
109
110     int ship_lengths[] = {4, 3, 3, 2, 2, 2, 1, 1, 1, 1};
111     int num_ships = 10;
112
113     for (int ship_idx = 0; ship_idx < num_ships; ship_idx++) {
114         int length = ship_lengths[ship_idx];
115         int placed = 0;
116         int attempts = 0;
117
118         while (!placed && attempts < MAX_ATTEMPTS) {
119             int row = rand() % BOARD_SIZE;
120             int col = rand() % BOARD_SIZE;
121             int horizontal = rand() % 2;
122
123             if (is_valid_placement(board, row, col, length, horizontal)) {
124                 place_ship(board, row, col, length, horizontal);
125                 placed = 1;
126             }
127
128             attempts++;
129         }
130
131         if (!placed) {
132             ship_idx--;
133             if (ship_idx < 0) ship_idx = 0;
134             init_board(board);
135         }
136     }
137 }
138
139 int is_ship_sunk(Board *board, int row, int col) {
140     if (board->cells[row][col] != HIT) {
141         return 0;
142     }
143
144     int col_start = col;
145     int col_end = col;
146
147     while (col_start > 0 && (board->cells[row][col_start - 1] == SHIP ||
148                             board->cells[row][col_start - 1] == HIT)) {
149         col_start--;
150     }
151
152     while (col_end < BOARD_SIZE - 1 && (board->cells[row][col_end + 1] == SHIP ||
153                                         board->cells[row][col_end + 1] == HIT)) {
154         col_end++;
155     }
156

```

```

157     int horizontal_intact = 0;
158     for (int c = col_start; c <= col_end; c++) {
159         if (board->cells[row][c] == SHIP) {
160             horizontal_intact = 1;
161             break;
162         }
163     }
164
165     if (!horizontal_intact && col_start < col_end) {
166         return 1;
167     }
168
169     int row_start = row;
170     int row_end = row;
171
172     while (row_start > 0 && (board->cells[row_start - 1][col] == SHIP ||
173         board->cells[row_start - 1][col] == HIT)) {
174         row_start--;
175     }
176
177     while (row_end < BOARD_SIZE - 1 && (board->cells[row_end + 1][col] == SHIP ||
178         board->cells[row_end + 1][col] == HIT)) {
179         row_end++;
180     }
181
182     int vertical_intact = 0;
183     for (int r = row_start; r <= row_end; r++) {
184         if (board->cells[r][col] == SHIP) {
185             vertical_intact = 1;
186             break;
187         }
188     }
189
190     if (!vertical_intact && row_start < row_end) {
191         return 1;
192     }
193
194     if (!horizontal_intact && col_start == col_end && !vertical_intact &&
195         row_start == row_end) {
196         return 1;
197     }
198
199     return 0;
200 }
201
202 int process_shot(Board *target_board, int row, int col) {
203     CellState cell = target_board->cells[row][col];
204
205     if (cell == EMPTY) {
206         target_board->cells[row][col] = MISS;
207         return 0;

```

```

208     } else if (cell == SHIP) {
209         target_board->cells[row][col] = HIT;
210         if (is_ship_sunk(target_board, row, col)) {
211             return 2;
212         }
213         return 1;
214     } else if (cell == HIT || cell == MISS) {
215         return -1;
216     }
217
218     return 0;
219 }
220
221 int all_ships_sunk(Board *board) {
222     for (int i = 0; i < BOARD_SIZE; i++) {
223         for (int j = 0; j < BOARD_SIZE; j++) {
224             if (board->cells[i][j] == SHIP) {
225                 return 0;
226             }
227         }
228     }
229     return 1;
230 }
231
232 char cell_to_char(CellState state, int reveal_ships) {
233     switch (state) {
234         case EMPTY:
235             return '.';
236         case SHIP:
237             return reveal_ships ? 'S' : '.';
238         case HIT:
239             return 'X';
240         case MISS:
241             return '0';
242         default:
243             return '?';
244     }
245 }

```

Листинг 12: *Реализация игровой логики*

```

1  #include <signal.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #include <unistd.h>
6
7  #include "common.h"
8  #include "game_logic.h"
9  #include "server.h"

```

```

10 #include "stats.h"
11 #include "sync_init.h"
12
13 static int keep_running = 1;
14
15 void signal_handler(int sig) {
16     (void)sig;
17     keep_running = 0;
18 }
19
20 int main(int argc, char *argv[]) {
21     (void)argc;
22     (void)argv;
23
24     printf("
25     printf(" BATTLESHIP - SERVER \n");
26
27     printf("
28     os_signal_register_int(signal_handler);
29
30     OSMmapHandle state_handle;
31     ServerState *state = NULL;
32
33     if (!server_init(&state, state_handle)) {
34         fprintf(stderr, "Error initializing server\n");
35         return 1;
36     }
37
38     printf("Initializing synchronization primitives...\n");
39     if (!init_server_mutexes(state)) {
40         fprintf(stderr, "Error initializing mutexes\n");
41         server_cleanup(state, state_handle);
42         return 1;
43     }
44
45     printf("Synchronization primitives initialized\n\n");
46
47     stats_init();
48     printf("Statistics system initialized\n\n");
49
50     printf("Server is running and waiting for players...\n");
51     printf("Clients can connect\n\n");
52
53     int last_game_count = 0;
54
55     while (keep_running) {
56         os_usleep(500000);
57
58         OSMmapHandle handle = mmap_open(STATE_FILE, sizeof(ServerState));

```

```

59     if (handle.addr == NULL) continue;
60
61     mmap_read(handle.addr, state, sizeof(ServerState));
62
63     os_mutex_lock(&state->state_mutex);
64
65     for (int i = 0; i < state->game_count; i++) {
66         GameState *game = &state->games[i];
67         os_mutex_lock(&game->game_mutex);
68
69         for (int shooter_idx = 0; shooter_idx < 2; shooter_idx++) {
70             if (game->last_shots[shooter_idx].active &&
71                 game->last_shots[shooter_idx].result == -2) {
72                 int row = game->last_shots[shooter_idx].row;
73                 int col = game->last_shots[shooter_idx].col;
74
75                 printf("\n[GAME: %s] Processing shot from player %d\n",
76                     game->game_name, shooter_idx);
77
78                 if (row < 0 || row >= BOARD_SIZE || col < 0 || col >= BOARD_SIZE) {
79                     printf("Error: coordinates out of bounds\n");
80                     game->last_shots[shooter_idx].result = -1;
81                 } else {
82                     int result = process_server_shot(game, shooter_idx, row, col);
83                     game->last_shots[shooter_idx].result = result;
84
85                     switch (result) {
86                         case 0:
87                             printf("MISS [%d,%d]\n", row, col);
88                             break;
89                         case 1:
90                             printf("HIT [%d,%d]\n", row, col);
91                             break;
92                         case 2:
93                             printf("SUNK [%d,%d]\n", row, col);
94                             break;
95                         case 3:
96                             printf("*** VICTORY! Player %d won!\n", shooter_idx);
97                             game->status = GAME_FINISHED;
98                             game->winner_idx = shooter_idx;
99                             break;
100                        case -1:
101                            printf("ERROR during processing\n");
102                            break;
103                    }
104
105                    game->last_update_version++;
106                    os_condvar_broadcast(&game->shot_changed);
107                    printf("Result sent (version %d)\n", game->last_update_version);
108                }
109            }

```

```

110     }
111
112     if (game->status == GAME_FINISHED && game->winner_idx >= 0 &&
113         !game->stats_updated) {
114         printf("\n[GAME %s] Finished!\n", game->game_name);
115         printf(" Winner: %s\n", game->players[game->winner_idx].login);
116         printf(" Loser: %s\n", game->players[1 - game->winner_idx].login);
117
118         update_player_stats(game->players[game->winner_idx].login, 1, 0);
119         int loser_idx = 1 - game->winner_idx;
120         update_player_stats(game->players[loser_idx].login, 0, 1);
121
122         game->stats_updated = 1;
123         game->last_shots[0].active = 0;
124         game->last_shots[1].active = 0;
125         os_condvar_broadcast(&game->shot_changed);
126     }
127
128     os_mutex_unlock(&game->game_mutex);
129 }
130
131 os_mutex_unlock(&state->state_mutex);
132 mmap_close(handle, sizeof(ServerState));
133
134 if (state->game_count != last_game_count) {
135     printf("\n[SERVER] Active games: %d/%d\n", state->game_count, MAX_GAMES);
136     last_game_count = state->game_count;
137 }
138 }
139
140 printf("\n[SERVER] Shutting down...\n");
141
142 for (int i = 0; i < state->game_count; i++) {
143     GameState *game = &state->games[i];
144
145     os_mutex_lock(&game->game_mutex);
146     game->status = GAME_FINISHED;
147     game->winner_idx = -1;
148     os_condvar_broadcast(&game->shot_changed);
149     os_mutex_unlock(&game->game_mutex);
150 }
151
152 os_usleep(5000000);
153 cleanup_server_mutexes(state);
154 server_cleanup(state, state_handle);
155 printf("Server stopped\n");
156
157 return 0;
158 }

```

Листинг 13: *Реализация сервера*

```

1  #include <errno.h>
2  #include <pthread.h>
3  #include <time.h>
4
5  #include "common.h"
6  #include "game_logic.h"
7  #include "server.h"
8  #include "stats.h"
9
10 #define SHOT_TIMEOUT 300
11
12 static char player_login[MAX_LOGIN];
13 static char current_game_id[MAX_GAME_ID];
14 static int player_index = -1;
15
16 static ServerState *get_server_state(OSMmapHandle *handle) {
17     *handle = mmap_open("server_state.mmap", sizeof(ServerState));
18     return (ServerState *)handle->addr;
19 }
20
21 static void clear_input_buffer(void) {
22     int c;
23     while ((c = getchar()) != '\n' && c != EOF) {
24     }
25 }
26
27 static int login_menu() {
28     printf("\n===== \n");
29     printf("=  BATTLESHIP - CLIENT                = \n");
30     printf("===== \n \n");
31
32     printf("Enter your login (up to 63 characters): ");
33     if (fgets(player_login, sizeof(player_login), stdin) == NULL) {
34         return 0;
35     }
36     player_login[strcspn(player_login, "\n")] = 0;
37
38     if (strlen(player_login) == 0) {
39         printf("Error: login cannot be empty \n");
40         return 0;
41     }
42
43     printf("Welcome, %s! \n", player_login);
44     stats_init();
45
46     return 1;
47 }
48
49 static int main_menu() {
50     printf("\n===== MAIN MENU ===== \n");
51     printf("1. Create new game \n");

```

```

52     printf("2. Join a game\n");
53     printf("3. View my statistics\n");
54     printf("4. Exit\n");
55     printf("Choose action (1-4): ");
56
57     char choice[10];
58     if (fgets(choice, sizeof(choice), stdin) == NULL) {
59         return 4;
60     }
61
62     return atoi(choice);
63 }
64
65 static int create_game_dialog() {
66     char game_name[MAX_GAME_NAME];
67
68     printf("\nEnter name of new game: ");
69     if (fgets(game_name, sizeof(game_name), stdin) == NULL) {
70         return 0;
71     }
72     game_name[strcspn(game_name, "\n")] = 0;
73
74     if (strlen(game_name) == 0) {
75         printf("Error: game name cannot be empty\n");
76         return 0;
77     }
78
79     OSMapHandle fd;
80     ServerState *state = get_server_state(&fd);
81     if (state == NULL) {
82         fprintf(stderr, "Error: could not connect to server\n");
83         return 0;
84     }
85
86     if (create_game(state, game_name, player_login, current_game_id)) {
87         mmap_close(fd, sizeof(ServerState));
88         player_index = 0;
89         printf("Game created! ID: %s\n", current_game_id);
90         printf("Waiting for second player...\n");
91         return 1;
92     }
93
94     mmap_close(fd, sizeof(ServerState));
95     return 0;
96 }
97
98 static int join_game_dialog() {
99     OSMapHandle fd;
100     ServerState *state = get_server_state(&fd);
101     if (state == NULL) {
102         fprintf(stderr, "Error: could not connect to server\n");

```

```

103     return 0;
104 }
105
106 list_games(state);
107
108 printf("\nEnter game ID to join (or empty to cancel): ");
109 char game_id[MAX_GAME_ID];
110 if (fgets(game_id, sizeof(game_id), stdin) == NULL) {
111     mmap_close(fd, sizeof(ServerState));
112     return 0;
113 }
114 game_id[strcspn(game_id, "\n")] = 0;
115
116 if (strlen(game_id) == 0) {
117     mmap_close(fd, sizeof(ServerState));
118     return 0;
119 }
120
121 if (join_game(state, game_id, player_login)) {
122     strcpy(current_game_id, game_id);
123     mmap_close(fd, sizeof(ServerState));
124     player_index = 1;
125     printf("Joined the game!\n");
126     return 1;
127 }
128
129 mmap_close(fd, sizeof(ServerState));
130 return 0;
131 }
132
133 static void setup_ships(Board *board) {
134     printf("\n===== SHIP PLACEMENT =====\n");
135     printf("Generating random ship placement...\n");
136     printf("- 1 four-deck ship (1x4)\n");
137     printf("- 2 three-deck ships (2x3)\n");
138     printf("- 3 two-deck ships (3x2)\n");
139     printf("- 4 one-deck ships (4x1)\n\n");
140
141     randomize_board(board);
142
143     printf("\n%s\n", "Ships placed randomly!");
144     printf("  0 1 2 3 4 5 6 7 8 9\n");
145     for (int i = 0; i < BOARD_SIZE; i++) {
146         printf(" %d ", i);
147         for (int j = 0; j < BOARD_SIZE; j++) {
148             printf("%c ", cell_to_char(board->cells[i][j], 1));
149         }
150         printf("\n");
151     }
152 }
153

```

```

154 static int get_opponent_index() { return (player_index == 0) ? 1 : 0; }
155
156 static void display_boards(const Board *my_board, const Board *my_shots) {
157     printf("\nYour board:           Your shots:\n");
158     printf("  0 1 2 3 4 5 6 7 8 9    0 1 2 3 4 5 6 7 8 9\n");
159
160     for (int i = 0; i < BOARD_SIZE; i++) {
161         printf(" %d ", i);
162         for (int j = 0; j < BOARD_SIZE; j++) {
163             printf("%c ", cell_to_char(my_board->cells[i][j], 1));
164         }
165         printf(" %d ", i);
166         for (int j = 0; j < BOARD_SIZE; j++) {
167             printf("%c ", cell_to_char(my_shots->cells[i][j], 0));
168         }
169         printf("\n");
170     }
171 }
172
173 int wait_for_shot_processing(GameState *game, int player_idx, int max_seconds) {
174     printf("...Waiting for server to process shot...\n");
175
176     os_mutex_lock(&game->game_mutex);
177
178     while (game->last_shots[player_idx].result == -2) {
179         int wait_result = os_condvar_timedwait(
180             &game->shot_changed, &game->game_mutex, max_seconds * 1000);
181
182         if (wait_result == OS_WAIT_TIMEOUT) {
183             printf("Timeout: server did not respond in %d seconds\n", max_seconds);
184             os_mutex_unlock(&game->game_mutex);
185             return 0;
186         }
187
188         if (wait_result == OS_WAIT_ERROR) {
189             printf("Wait error\n");
190             os_mutex_unlock(&game->game_mutex);
191             return 0;
192         }
193     }
194
195     os_mutex_unlock(&game->game_mutex);
196
197     return 1;
198 }
199
200 static void play_game() {
201     printf("\n===== GAME START =====\n");
202     OSMmapHandle fd;
203     ServerState *state = get_server_state(&fd);
204

```

```

205     if (state == NULL) {
206         fprintf(stderr, "Error: could not connect to server\n");
207         return;
208     }
209
210     GameState *game = get_game(state, current_game_id);
211     if (game == NULL) {
212         fprintf(stderr, "Error: game not found\n");
213         mmap_close(fd, sizeof(ServerState));
214         return;
215     }
216
217     int opponent_idx = get_opponent_index();
218     int game_over = 0;
219     int last_opponent_shot_version = -1;
220
221     while (!game_over) {
222         mmap_read(state, state, sizeof(ServerState));
223         game = get_game(state, current_game_id);
224
225         if (game == NULL) {
226             fprintf(stderr, "Error: game lost\n");
227             break;
228         }
229
230         if (game->last_shots[opponent_idx].result == 3) {
231             printf("\n~~~ YOU LOST! ~~~\n");
232             game->status = GAME_FINISHED;
233             game->winner_idx = opponent_idx;
234             game_over = 1;
235             break;
236         }
237
238         Board my_board = game->players[player_index].board;
239         Board my_shots = game->players[player_index].shots;
240
241         printf("\n===== CURRENT STATUS =====\n");
242
243         display_boards(&my_board, &my_shots);
244
245         if (game->current_turn == player_index) {
246             printf("\n=== YOUR TURN ===\n");
247             printf("Enter shot coordinates (row column): ");
248
249             int row, col;
250             if (scanf("%d %d", &row, &col) != 2) {
251                 clear_input_buffer();
252                 printf("Error: enter two numbers!\n");
253                 continue;
254             }
255

```

```

256     clear_input_buffer();
257
258     if (row < 0 || row >= BOARD_SIZE || col < 0 || col >= BOARD_SIZE) {
259         printf("Error: coordinates out of bounds (0-%d)\n", BOARD_SIZE - 1);
260         continue;
261     }
262
263     if (my_shots.cells[row][col] != EMPTY) {
264         printf("Error: you already shot here!\n");
265         continue;
266     }
267
268     printf("...Sending shot to server...\n");
269
270     os_mutex_lock(&game->game_mutex);
271
272     game->last_shots[player_index].active = 1;
273     game->last_shots[player_index].row = row;
274     game->last_shots[player_index].col = col;
275     game->last_shots[player_index].result = -2;
276     game->last_shots[player_index].processed_by_opponent = 0;
277     game->last_update_version++;
278
279     for (int i = 0; i < state->game_count; i++) {
280         if (strcmp(state->games[i].game_id, current_game_id) == 0) {
281             state->games[i] = *game;
282             mmap_write(&state->games[i], &state->games[i], sizeof(GameState));
283             break;
284         }
285     }
286
287     mmap_write(state, state, sizeof(ServerState));
288
289     os_mutex_unlock(&game->game_mutex);
290
291     if (!wait_for_shot_processing(game, player_index, 10)) {
292         printf("Error: server did not process shot\n");
293         continue;
294     }
295
296     mmap_read(state, state, sizeof(ServerState));
297     game = get_game(state, current_game_id);
298     if (game == NULL) break;
299
300     int result = game->last_shots[player_index].result;
301
302     if (result == -1) {
303         printf("Error: invalid shot\n");
304     } else if (result == 0) {
305         printf("MISS! Turn goes to opponent.\n");
306     } else if (result == 1) {

```

```

307     printf("HIT! You shoot again!\n");
308 } else if (result == 2) {
309     printf("SUNK! You shoot again!\n");
310 } else if (result == 3) {
311     printf("*** YOU WON! ***\n");
312     game_over = 1;
313     break;
314 }
315
316 os_mutex_lock(&game->game_mutex);
317 game->last_shots[player_index].active = 0;
318 os_mutex_unlock(&game->game_mutex);
319
320 for (int i = 0; i < state->game_count; i++) {
321     if (strcmp(state->games[i].game_id, current_game_id) == 0) {
322         state->games[i] = *game;
323         mmap_write(&state->games[i], &state->games[i], sizeof(GameState));
324         break;
325     }
326 }
327 mmap_write(state, state, sizeof(ServerState));
328
329 } else {
330     printf("\n=== WAITING FOR OPPONENT'S TURN ===\n");
331
332     int opponent_made_move = 0;
333
334     os_mutex_lock(&game->game_mutex);
335
336     while (game->current_turn == opponent_idx &&
337           game->status != GAME_FINISHED && !opponent_made_move) {
338         int wait_result = os_condvar_timedwait(&game->shot_changed,
339                                               &game->game_mutex, 120000);
340
341         if (wait_result == OS_WAIT_TIMEOUT) {
342             printf("Opponent is taking too long\n");
343             os_mutex_unlock(&game->game_mutex);
344             break;
345         }
346
347         if (wait_result == OS_WAIT_ERROR) {
348             os_mutex_unlock(&game->game_mutex);
349             break;
350         }
351
352         if (game->last_update_version != last_opponent_shot_version) {
353             opponent_made_move = 1;
354         }
355     }
356
357     os_mutex_unlock(&game->game_mutex);

```

```

358
359     mmap_read(state, state, sizeof(ServerState));
360     game = get_game(state, current_game_id);
361     if (game == NULL) break;
362
363     my_board = game->players[player_index].board;
364     my_shots = game->players[player_index].shots;
365
366     if (game->last_shots[opponent_idx].result >= -1 &&
367         last_opponent_shot_version != game->last_update_version) {
368         last_opponent_shot_version = game->last_update_version;
369
370         printf("\n*** OPPONENT SHOT! ***\n");
371         int row = game->last_shots[opponent_idx].row;
372         int col = game->last_shots[opponent_idx].col;
373         int result = game->last_shots[opponent_idx].result;
374
375         printf("Opponent shot [%d,%d]: ", row, col);
376
377         if (result == 0) {
378             printf("MISS\n");
379             my_board.cells[row][col] = MISS;
380         } else if (result == 1) {
381             printf("HIT your cell!\n");
382             my_board.cells[row][col] = HIT;
383         } else if (result == 2) {
384             printf("SUNK your ship!\n");
385             my_board.cells[row][col] = HIT;
386         } else if (result == 3) {
387             printf("~~~YOU LOST!~~~\n");
388         }
389
390         game->last_shots[opponent_idx].processed_by_opponent = 1;
391
392         for (int i = 0; i < state->game_count; i++) {
393             if (strcmp(state->games[i].game_id, current_game_id) == 0) {
394                 state->games[i] = *game;
395                 mmap_write(&state->games[i], &state->games[i], sizeof(GameState));
396                 break;
397             }
398         }
399         mmap_write(state, state, sizeof(ServerState));
400     }
401 }
402 }
403
404 mmap_close(fd, sizeof(ServerState));
405
406 stats_init();
407 printf("\nStatistics updated from server\n");
408 clear_input_buffer;

```

```

409 }
410
411 int main() {
412     init_random();
413
414     if (!login_menu()) {
415         printf("Error logging in\n");
416         return 1;
417     }
418
419     int running = 1;
420
421     while (running) {
422         int choice = main_menu();
423
424         switch (choice) {
425             case 1:
426                 if (create_game_dialog()) {
427                     OSMmapHandle fd;
428                     ServerState *state = get_server_state(&fd);
429                     GameState *game = get_game(state, current_game_id);
430
431                     setup_ships(&game->players[player_index].board);
432                     game->players[player_index].ready = 1;
433                     mmap_write(game, game, sizeof(GameState));
434
435                     printf("\nWaiting for second player...\n");
436                     printf("(When second player connects, ship placement will begin)\n");
437
438                     int waiting = 1;
439                     while (waiting && game->player_count < 2) {
440                         os_usleep(500000);
441                         mmap_read(state, state, sizeof(ServerState));
442                         game = get_game(state, current_game_id);
443                         printf(".");
444                         fflush(stdout);
445                     }
446                     printf("\nSecond player connected!\n");
447
448                     mmap_close(fd, sizeof(ServerState));
449                     play_game();
450                 }
451                 break;
452
453             case 2:
454                 if (join_game_dialog()) {
455                     OSMmapHandle fd;
456                     ServerState *state = get_server_state(&fd);
457                     GameState *game = get_game(state, current_game_id);
458
459                     setup_ships(&game->players[player_index].board);

```

```

460     game->players[player_index].ready = 1;
461     mmap_write(game, game, sizeof(GameState));
462
463     printf("Waiting for first player...\n");
464     int waiting = 1;
465     while (waiting && !game->players[0].ready) {
466         os_usleep(500000);
467         mmap_read(state, state, sizeof(ServerState));
468         game = get_game(state, current_game_id);
469     }
470     printf("Both players ready! Game starts!\n");
471
472     game->status = GAME_RUNNING;
473     mmap_write(game, game, sizeof(GameState));
474     mmap_close(fd, sizeof(ServerState));
475     play_game();
476 }
477 break;
478
479 case 3:
480     stats_init();
481     display_player_stats(player_login);
482     break;
483
484 case 4:
485     running = 0;
486     printf("Goodbye!\n");
487     break;
488
489 default:
490     printf("Invalid choice\n");
491 }
492 }
493
494 return 0;
495 }

```

Листинг 14: *Реализация клиента*

```

1 1641 17:21:17.007127 execve("./battleship_server", ["./battleship_server"],
    0x7ffeb1584780 /* 36 vars */) = 0
2 1641 17:21:17.008394 brk(NULL) = 0x55fb185ad000
3 1641 17:21:17.008810 arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc338147f0) = -1
    EINVAL (Invalid argument)
4 1641 17:21:17.009162 mmap(NULL, 8192, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f2d20789000
5 1641 17:21:17.009551 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such
    file or directory)
6 1641 17:21:17.010077 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC)
    = 3

```

```

7 1641 17:21:17.010381 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=30404,
...}, AT_EMPTY_PATH) = 0
8 1641 17:21:17.010641 mmap(NULL, 30404, PROT_READ, MAP_PRIVATE, 3, 0) =
0x7f2d20781000
9 1641 17:21:17.010965 close(3) = 0
10 1641 17:21:17.011285 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6",
O_RDONLY|O_CLOEXEC) = 3
11 1641 17:21:17.011484 read(3,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0"..., 832)
= 832
12 1641 17:21:17.011623 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"..., 784,
64) = 784
13 1641 17:21:17.011759 pread64(3, "\4\0\0\0 \
0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0"..., 48, 848) = 48
14 1641 17:21:17.011902 pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\00{\f\225\\=\201\327\312\301P\32$\230\266\235"...,
68, 896) = 68
15 1641 17:21:17.012337 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400,
...}, AT_EMPTY_PATH) = 0
16 1641 17:21:17.012758 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"..., 784,
64) = 784
17 1641 17:21:17.013061 mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE,
3, 0) = 0x7f2d20558000
18 1641 17:21:17.013412 mprotect(0x7f2d20580000, 2023424, PROT_NONE) = 0
19 1641 17:21:17.013810 mmap(0x7f2d20580000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f2d20580000
20 1641 17:21:17.014219 mmap(0x7f2d20715000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f2d20715000
21 1641 17:21:17.014457 mmap(0x7f2d2076e000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f2d2076e000
22 1641 17:21:17.014723 mmap(0x7f2d20774000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2d20774000
23 1641 17:21:17.015012 close(3) = 0
24 1641 17:21:17.015251 mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f2d20555000
25 1641 17:21:17.015528 arch_prctl(ARCH_SET_FS, 0x7f2d20555740) = 0
26 1641 17:21:17.015878 set_tid_address(0x7f2d20555a10) = 1641
27 1641 17:21:17.016178 set_robust_list(0x7f2d20555a20, 24) = 0
28 1641 17:21:17.016381 rseq(0x7f2d205560e0, 0x20, 0, 0x53053053) = 0
29 1641 17:21:17.016642 mprotect(0x7f2d2076e000, 16384, PROT_READ) = 0
30 1641 17:21:17.016912 mprotect(0x55fb0885d000, 4096, PROT_READ) = 0
31 1641 17:21:17.017297 mprotect(0x7f2d207c3000, 8192, PROT_READ) = 0
32 1641 17:21:17.017593 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
33 1641 17:21:17.017846 munmap(0x7f2d20781000, 30404) = 0
34 1641 17:21:17.018229 newfstatat(1, "", {st_mode=S_IFCHR|0620,
st_rdev=makedev(0x88, 0x4), ...}, AT_EMPTY_PATH) = 0
35 1641 17:21:17.018587 getrandom("\x4e\x23\xf9\xe2\x0d\xd9\x2f\xca", 8,
GRND_NONBLOCK) = 8

```

[illegible]

```

70 1641 17:21:17.540043 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
71 1641 17:21:18.041347 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
72 1641 17:21:18.041904 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
73 1641 17:21:18.042385 munmap(0x7f2d20543000, 32864) = 0
74 1641 17:21:18.042727 close(4) = 0
75 1641 17:21:18.042911 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
76 1641 17:21:16.789523 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
77 1641 17:21:16.789834 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
78 1641 17:21:16.790177 munmap(0x7f2d20543000, 32864) = 0
79 1641 17:21:16.790406 close(4) = 0
80 1641 17:21:16.790625 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
81 1641 17:21:17.311394 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
82 1641 17:21:17.312119 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
83 1641 17:21:17.312815 munmap(0x7f2d20543000, 32864) = 0
84 1641 17:21:17.313233 close(4) = 0
85 1641 17:21:17.313660 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
86 1641 17:21:17.825330 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
87 1641 17:21:17.825883 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
88 1641 17:21:17.826366 munmap(0x7f2d20543000, 32864) = 0
89 1641 17:21:17.827114 close(4) = 0
90 1641 17:21:17.827486 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
91 1641 17:21:18.330641 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
92 1641 17:21:18.331094 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
93 1641 17:21:18.331493 munmap(0x7f2d20543000, 32864) = 0
94 1641 17:21:18.331797 close(4) = 0
95 1641 17:21:18.332082 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
96 1641 17:21:18.848632 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
97 1641 17:21:18.849399 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
98 1641 17:21:18.849918 munmap(0x7f2d20543000, 32864) = 0
99 1641 17:21:18.850223 close(4) = 0
100 1641 17:21:18.850570 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
101 1641 17:21:19.387723 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
102 1641 17:21:19.388456 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
103 1641 17:21:19.388929 munmap(0x7f2d20543000, 32864) = 0
104 1641 17:21:19.389362 close(4) = 0
105 1641 17:21:19.389734 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0

```

```

106 1641 17:21:19.908950 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
107 1641 17:21:19.909383 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
108 1641 17:21:19.909700 munmap(0x7f2d20543000, 32864) = 0
109 1641 17:21:19.910014 close(4) = 0
110 1641 17:21:19.910291 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
111 1641 17:21:20.429200 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
112 1641 17:21:20.429760 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
113 1641 17:21:20.430098 munmap(0x7f2d20543000, 32864) = 0
114 1641 17:21:20.430471 close(4) = 0
115 1641 17:21:20.430897 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
116 1641 17:21:20.942708 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
117 1641 17:21:20.943495 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
118 1641 17:21:20.943928 munmap(0x7f2d20543000, 32864) = 0
119 1641 17:21:20.944313 close(4) = 0
120 1641 17:21:20.944696 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
121 1641 17:21:21.448271 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
122 1641 17:21:21.448770 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
123 1641 17:21:21.449262 munmap(0x7f2d20543000, 32864) = 0
124 1641 17:21:21.449562 close(4) = 0
125 1641 17:21:21.449974 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
126 1641 17:21:21.951458 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
127 1641 17:21:21.952119 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
128 1641 17:21:21.952726 munmap(0x7f2d20543000, 32864) = 0
129 1641 17:21:21.953289 close(4) = 0
130 1641 17:21:21.953815 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
131 1641 17:21:22.461644 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
132 1641 17:21:22.462256 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
133 1641 17:21:22.462589 munmap(0x7f2d20543000, 32864) = 0
134 1641 17:21:22.462971 close(4) = 0
135 1641 17:21:22.463390 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
136 1641 17:21:22.965734 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
137 1641 17:21:22.966372 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
138 1641 17:21:22.966816 munmap(0x7f2d20543000, 32864) = 0
139 1641 17:21:22.967297 close(4) = 0
140 1641 17:21:22.967761 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
141 1641 17:21:23.474602 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
142 1641 17:21:23.475217 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)

```

```

    = 0x7f2d20543000
143 1641 17:21:23.475806 munmap(0x7f2d20543000, 32864) = 0
144 1641 17:21:23.476368 close(4) = 0
145 1641 17:21:23.476687 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
146 1641 17:21:23.979087 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
147 1641 17:21:23.979808 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
148 1641 17:21:23.980271 munmap(0x7f2d20543000, 32864) = 0
149 1641 17:21:23.980670 close(4) = 0
150 1641 17:21:23.981117 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
151 1641 17:21:24.483085 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
152 1641 17:21:24.483777 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
153 1641 17:21:24.484212 munmap(0x7f2d20543000, 32864) = 0
154 1641 17:21:24.484539 close(4) = 0
155 1641 17:21:24.484918 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
156 1641 17:21:24.987757 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
157 1641 17:21:24.988230 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
158 1641 17:21:24.988642 munmap(0x7f2d20543000, 32864) = 0
159 1641 17:21:24.989044 close(4) = 0
160 1641 17:21:24.989342 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
161 1641 17:21:25.490511 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
162 1641 17:21:25.490918 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
163 1641 17:21:25.491265 munmap(0x7f2d20543000, 32864) = 0
164 1641 17:21:25.491569 close(4) = 0
165 1641 17:21:25.491908 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
166 1641 17:21:25.992623 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
167 1641 17:21:25.993135 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
168 1641 17:21:25.993694 munmap(0x7f2d20543000, 32864) = 0
169 1641 17:21:25.994216 close(4) = 0
170 1641 17:21:25.994554 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
171 1641 17:21:26.502746 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
172 1641 17:21:26.503376 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
173 1641 17:21:26.503682 munmap(0x7f2d20543000, 32864) = 0
174 1641 17:21:26.503984 close(4) = 0
175 1641 17:21:26.504340 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
176 1641 17:21:27.008111 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
177 1641 17:21:27.008571 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
178 1641 17:21:27.009111 munmap(0x7f2d20543000, 32864) = 0

```

```

179 1641 17:21:27.009492 close(4) = 0
180 1641 17:21:27.009922 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
181 1641 17:21:27.511954 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
182 1641 17:21:27.512522 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
183 1641 17:21:27.512963 munmap(0x7f2d20543000, 32864) = 0
184 1641 17:21:27.513338 close(4) = 0
185 1641 17:21:27.513577 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
186 1641 17:21:28.016515 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
187 1641 17:21:28.017165 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
188 1641 17:21:28.017638 munmap(0x7f2d20543000, 32864) = 0
189 1641 17:21:28.017952 close(4) = 0
190 1641 17:21:28.018299 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
191 1641 17:21:28.521974 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
192 1641 17:21:28.522631 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
193 1641 17:21:28.523245 munmap(0x7f2d20543000, 32864) = 0
194 1641 17:21:28.523863 close(4) = 0
195 1641 17:21:28.524428 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
196 1641 17:21:29.026581 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
197 1641 17:21:29.027379 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
198 1641 17:21:29.027757 munmap(0x7f2d20543000, 32864) = 0
199 1641 17:21:29.028179 close(4) = 0
200 1641 17:21:29.028576 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
201 1641 17:21:29.541687 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
202 1641 17:21:29.542433 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
203 1641 17:21:29.543009 munmap(0x7f2d20543000, 32864) = 0
204 1641 17:21:29.543512 close(4) = 0
205 1641 17:21:29.544043 write(1, "\n", 1) = 1
206 1641 17:21:29.544539 write(1, "[SERVER] Active games: 1/16\n", 28) = 28
207 1641 17:21:29.544984 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
208 1641 17:21:30.051489 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
209 1641 17:21:30.052038 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
210 1641 17:21:30.052629 munmap(0x7f2d20543000, 32864) = 0
211 1641 17:21:30.053215 close(4) = 0
212 1641 17:21:30.053556 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
213 1641 17:21:30.561665 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
214 1641 17:21:30.562034 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
215 1641 17:21:30.562368 munmap(0x7f2d20543000, 32864) = 0

```

```
216 1641 17:21:30.562664 close(4) = 0
217 1641 17:21:30.563014 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
218 1641 17:21:31.071345 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
219 1641 17:21:31.072075 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
220 1641 17:21:31.072404 munmap(0x7f2d20543000, 32864) = 0
221 1641 17:21:31.072695 close(4) = 0
222 1641 17:21:31.073051 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
223 1641 17:21:31.578975 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
224 1641 17:21:31.579431 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
225 1641 17:21:31.579930 munmap(0x7f2d20543000, 32864) = 0
226 1641 17:21:31.580255 close(4) = 0
227 1641 17:21:31.580641 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
228 1641 17:21:32.084473 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
229 1641 17:21:32.085275 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
230 1641 17:21:32.086024 munmap(0x7f2d20543000, 32864) = 0
231 1641 17:21:32.086528 close(4) = 0
232 1641 17:21:32.086880 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
233 1641 17:21:32.588443 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
234 1641 17:21:32.589094 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
235 1641 17:21:32.589689 munmap(0x7f2d20543000, 32864) = 0
236 1641 17:21:32.590171 close(4) = 0
237 1641 17:21:32.590556 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
238 1641 17:21:33.127229 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
239 1641 17:21:33.127889 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
240 1641 17:21:33.128278 munmap(0x7f2d20543000, 32864) = 0
241 1641 17:21:33.128618 close(4) = 0
242 1641 17:21:33.128895 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
243 1641 17:21:33.634332 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
244 1641 17:21:33.635087 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
245 1641 17:21:33.635531 munmap(0x7f2d20543000, 32864) = 0
246 1641 17:21:33.635950 close(4) = 0
247 1641 17:21:33.636290 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
248 1641 17:21:34.139844 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
249 1641 17:21:34.140743 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
250 1641 17:21:34.141208 munmap(0x7f2d20543000, 32864) = 0
251 1641 17:21:34.141640 close(4) = 0
252 1641 17:21:34.141928 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
```

```

    tv_nsec=500000000}, NULL) = 0
253 1641 17:21:34.652087 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
254 1641 17:21:34.652864 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
255 1641 17:21:34.653316 munmap(0x7f2d20543000, 32864) = 0
256 1641 17:21:34.653748 close(4) = 0
257 1641 17:21:34.654099 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
258 1641 17:21:35.157387 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
259 1641 17:21:35.157991 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
260 1641 17:21:35.158424 munmap(0x7f2d20543000, 32864) = 0
261 1641 17:21:35.158760 close(4) = 0
262 1641 17:21:35.159138 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
263 1641 17:21:35.661413 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
264 1641 17:21:35.661836 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
265 1641 17:21:35.662295 munmap(0x7f2d20543000, 32864) = 0
266 1641 17:21:35.662687 close(4) = 0
267 1641 17:21:35.662963 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
268 1641 17:21:36.164439 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
269 1641 17:21:36.165310 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
270 1641 17:21:36.166047 munmap(0x7f2d20543000, 32864) = 0
271 1641 17:21:36.166509 close(4) = 0
272 1641 17:21:36.166862 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
273 1641 17:21:36.669104 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
274 1641 17:21:36.669540 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
275 1641 17:21:36.669916 munmap(0x7f2d20543000, 32864) = 0
276 1641 17:21:36.670371 close(4) = 0
277 1641 17:21:36.670623 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
278 1641 17:21:36.930017 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
279 1641 17:21:36.930202 restart_syscall(<... resuming interrupted clock_nanosleep
    ...>) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
280 1641 17:21:36.930653 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
281 1641 17:21:36.930787 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
282 1641 17:21:36.930921 restart_syscall(<... resuming interrupted restart_syscall
    ...>) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
283 1641 17:21:37.047103 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
284 1641 17:21:37.047433 restart_syscall(<... resuming interrupted restart_syscall
    ...>) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
285 1641 17:21:37.110709 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
286 1641 17:21:37.110912 restart_syscall(<... resuming interrupted restart_syscall
    ...>) = 0
287 1641 17:21:37.172048 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
288 1641 17:21:37.172688 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)

```

```

    = 0x7f2d20543000
289 1641 17:21:37.173093 munmap(0x7f2d20543000, 32864) = 0
290 1641 17:21:37.173573 close(4) = 0
291 1641 17:21:37.173841 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
292 1641 17:21:37.680013 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
293 1641 17:21:37.680708 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
294 1641 17:21:37.681153 munmap(0x7f2d20543000, 32864) = 0
295 1641 17:21:37.681580 close(4) = 0
296 1641 17:21:37.681871 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
297 1641 17:21:38.182722 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
298 1641 17:21:38.183199 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
299 1641 17:21:38.183461 munmap(0x7f2d20543000, 32864) = 0
300 1641 17:21:38.183713 close(4) = 0
301 1641 17:21:38.183987 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
302 1641 17:21:38.687941 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
303 1641 17:21:38.688467 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
304 1641 17:21:38.688877 munmap(0x7f2d20543000, 32864) = 0
305 1641 17:21:38.689189 close(4) = 0
306 1641 17:21:38.689478 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
307 1641 17:21:39.213018 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
308 1641 17:21:39.213763 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
309 1641 17:21:39.214169 munmap(0x7f2d20543000, 32864) = 0
310 1641 17:21:39.214563 close(4) = 0
311 1641 17:21:39.214915 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
312 1641 17:21:39.735837 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
313 1641 17:21:39.736415 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
314 1641 17:21:39.736891 munmap(0x7f2d20543000, 32864) = 0
315 1641 17:21:39.737371 close(4) = 0
316 1641 17:21:39.737727 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
317 1641 17:21:40.238151 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
318 1641 17:21:40.238515 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
319 1641 17:21:40.238886 munmap(0x7f2d20543000, 32864) = 0
320 1641 17:21:40.239200 close(4) = 0
321 1641 17:21:40.239371 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
322 1641 17:21:40.760794 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
323 1641 17:21:40.761626 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
324 1641 17:21:40.762396 munmap(0x7f2d20543000, 32864) = 0

```

```
325 1641 17:21:40.762896 close(4) = 0
326 1641 17:21:40.763164 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
327 1641 17:21:41.275915 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
328 1641 17:21:41.276576 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
329 1641 17:21:41.277087 munmap(0x7f2d20543000, 32864) = 0
330 1641 17:21:41.277668 close(4) = 0
331 1641 17:21:41.277985 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
332 1641 17:21:41.796146 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
333 1641 17:21:41.796874 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
334 1641 17:21:41.797429 munmap(0x7f2d20543000, 32864) = 0
335 1641 17:21:41.797929 close(4) = 0
336 1641 17:21:41.798461 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
337 1641 17:21:42.298943 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
338 1641 17:21:42.299292 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
339 1641 17:21:42.299871 munmap(0x7f2d20543000, 32864) = 0
340 1641 17:21:42.300276 close(4) = 0
341 1641 17:21:42.300505 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
342 1641 17:21:42.819762 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
343 1641 17:21:42.820483 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
344 1641 17:21:42.820979 munmap(0x7f2d20543000, 32864) = 0
345 1641 17:21:42.821345 close(4) = 0
346 1641 17:21:42.821598 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
347 1641 17:21:43.345733 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
348 1641 17:21:43.346498 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
349 1641 17:21:43.347266 munmap(0x7f2d20543000, 32864) = 0
350 1641 17:21:43.347723 close(4) = 0
351 1641 17:21:43.347977 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
352 1641 17:21:43.871951 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
353 1641 17:21:43.872615 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
354 1641 17:21:43.873163 munmap(0x7f2d20543000, 32864) = 0
355 1641 17:21:43.873534 close(4) = 0
356 1641 17:21:43.873949 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
357 1641 17:21:44.374836 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
358 1641 17:21:44.375243 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
359 1641 17:21:44.375604 munmap(0x7f2d20543000, 32864) = 0
360 1641 17:21:44.375868 close(4) = 0
361 1641 17:21:44.376199 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
```

```

    tv_nsec=500000000}, NULL) = 0
362 1641 17:21:44.897970 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
363 1641 17:21:44.898323 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
364 1641 17:21:44.898891 munmap(0x7f2d20543000, 32864) = 0
365 1641 17:21:44.899405 close(4) = 0
366 1641 17:21:44.899781 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
367 1641 17:21:45.401225 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
368 1641 17:21:45.402086 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
369 1641 17:21:45.402567 munmap(0x7f2d20543000, 32864) = 0
370 1641 17:21:45.402876 close(4) = 0
371 1641 17:21:45.403171 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
372 1641 17:21:45.911403 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
373 1641 17:21:45.912036 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
374 1641 17:21:45.912447 munmap(0x7f2d20543000, 32864) = 0
375 1641 17:21:45.912819 close(4) = 0
376 1641 17:21:45.913203 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
377 1641 17:21:46.420887 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
378 1641 17:21:46.421319 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
379 1641 17:21:46.421732 munmap(0x7f2d20543000, 32864) = 0
380 1641 17:21:46.422077 close(4) = 0
381 1641 17:21:46.422462 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
382 1641 17:21:46.929458 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
383 1641 17:21:46.930205 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
384 1641 17:21:46.930511 munmap(0x7f2d20543000, 32864) = 0
385 1641 17:21:46.930786 close(4) = 0
386 1641 17:21:46.931066 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
387 1641 17:21:47.432477 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
388 1641 17:21:47.432961 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
389 1641 17:21:47.433372 munmap(0x7f2d20543000, 32864) = 0
390 1641 17:21:47.433763 close(4) = 0
391 1641 17:21:47.434037 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
392 1641 17:21:47.935757 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
393 1641 17:21:47.936349 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
394 1641 17:21:47.936760 munmap(0x7f2d20543000, 32864) = 0
395 1641 17:21:47.937152 close(4) = 0
396 1641 17:21:47.937660 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
397 1641 17:21:48.441150 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4

```

```

398 1641 17:21:48.441896 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
399 1641 17:21:48.442263 munmap(0x7f2d20543000, 32864) = 0
400 1641 17:21:48.442604 close(4) = 0
401 1641 17:21:48.443009 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
402 1641 17:21:47.081007 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
403 1641 17:21:47.081789 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
404 1641 17:21:47.082297 munmap(0x7f2d20543000, 32864) = 0
405 1641 17:21:47.082544 close(4) = 0
406 1641 17:21:47.082796 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
407 1641 17:21:47.584569 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
408 1641 17:21:47.585165 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
409 1641 17:21:47.585660 munmap(0x7f2d20543000, 32864) = 0
410 1641 17:21:47.585975 close(4) = 0
411 1641 17:21:47.586255 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
412 1641 17:21:48.089657 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
413 1641 17:21:48.090292 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
414 1641 17:21:48.090728 munmap(0x7f2d20543000, 32864) = 0
415 1641 17:21:48.091164 close(4) = 0
416 1641 17:21:48.091516 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
417 1641 17:21:48.594102 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
418 1641 17:21:48.594806 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
419 1641 17:21:48.595189 munmap(0x7f2d20543000, 32864) = 0
420 1641 17:21:48.595511 close(4) = 0
421 1641 17:21:48.595743 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
422 1641 17:21:49.098205 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
423 1641 17:21:49.098648 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
424 1641 17:21:49.099087 write(1, "\n", 1) = 1
425 1641 17:21:49.099634 write(1, "[GAME: lol] Processing shot from...", 42) = 42
426 1641 17:21:49.100125 write(1, "[SHOT] Player 0 shoots [0,2] -> "..., 63) = 63
427 1641 17:21:49.100435 write(1, "SUNK [0,2]\n", 11) = 11
428 1641 17:21:49.100687 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
429 1641 17:21:49.101070 write(1, "Result sent (version 2)\n", 24) = 24
430 1641 17:21:49.101524 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
431 1641 17:21:49.101950 munmap(0x7f2d20543000, 32864) = 0
432 1641 17:21:49.102243 close(4) = 0
433 1641 17:21:49.102584 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
434 1641 17:21:49.610013 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
435 1641 17:21:49.610327 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000

```

```
436 1641 17:21:49.610595 munmap(0x7f2d20543000, 32864) = 0
437 1641 17:21:49.610913 close(4) = 0
438 1641 17:21:49.611190 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
439 1641 17:21:50.111862 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
440 1641 17:21:50.112200 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
441 1641 17:21:50.112596 munmap(0x7f2d20543000, 32864) = 0
442 1641 17:21:50.113012 close(4) = 0
443 1641 17:21:50.113387 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
444 1641 17:21:50.615002 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
445 1641 17:21:50.615541 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
446 1641 17:21:50.616063 munmap(0x7f2d20543000, 32864) = 0
447 1641 17:21:50.616387 close(4) = 0
448 1641 17:21:50.616716 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
449 1641 17:21:51.118750 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
450 1641 17:21:51.119346 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
451 1641 17:21:51.119708 munmap(0x7f2d20543000, 32864) = 0
452 1641 17:21:51.120123 close(4) = 0
453 1641 17:21:51.120614 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
454 1641 17:21:51.622966 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
455 1641 17:21:51.623400 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
456 1641 17:21:51.624105 write(1, "\n", 1) = 1
457 1641 17:21:51.624719 write(1, "[GAME: lol] Processing shot from...", 42) = 42
458 1641 17:21:51.625153 write(1, "[SHOT] Player 0 shoots [1,5] -> ", 62) = 62
459 1641 17:21:51.625597 write(1, "HIT [1,5]\n", 10) = 10
460 1641 17:21:51.626101 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
461 1641 17:21:51.626681 write(1, "Result sent (version 4)\n", 24) = 24
462 1641 17:21:51.627132 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
463 1641 17:21:51.627547 munmap(0x7f2d20543000, 32864) = 0
464 1641 17:21:51.627845 close(4) = 0
465 1641 17:21:51.628227 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
466 1641 17:21:52.130647 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
467 1641 17:21:52.131161 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
468 1641 17:21:52.131644 munmap(0x7f2d20543000, 32864) = 0
469 1641 17:21:52.132172 close(4) = 0
470 1641 17:21:52.132505 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
471 1641 17:21:52.635481 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
472 1641 17:21:52.636027 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
473 1641 17:21:52.636336 munmap(0x7f2d20543000, 32864) = 0
474 1641 17:21:52.636787 close(4) = 0
```

```

475 1641 17:21:52.637070 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
476 1641 17:21:53.139184 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
477 1641 17:21:53.139806 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
478 1641 17:21:53.140161 write(1, "\n", 1) = 1
479 1641 17:21:53.140542 write(1, "[GAME: lol] Processing shot from...", 42) = 42
480 1641 17:21:53.141064 write(1, "[SHOT] Player 0 shoots [1,6] -> "..., 63) = 63
481 1641 17:21:53.141491 write(1, "SUNK [1,6]\n", 11) = 11
482 1641 17:21:53.141893 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
483 1641 17:21:53.142308 write(1, "Result sent (version 6)\n", 24) = 24
484 1641 17:21:53.142655 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
485 1641 17:21:53.143030 munmap(0x7f2d20543000, 32864) = 0
486 1641 17:21:53.143340 close(4) = 0
487 1641 17:21:53.143636 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
488 1641 17:21:53.663262 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
489 1641 17:21:53.663951 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
490 1641 17:21:53.664345 munmap(0x7f2d20543000, 32864) = 0
491 1641 17:21:53.664660 close(4) = 0
492 1641 17:21:53.664880 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
493 1641 17:21:54.166836 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
494 1641 17:21:54.167418 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
495 1641 17:21:54.167789 munmap(0x7f2d20543000, 32864) = 0
496 1641 17:21:54.168063 close(4) = 0
497 1641 17:21:54.168529 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
498 1641 17:21:54.671250 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
499 1641 17:21:54.671646 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
500 1641 17:21:54.672125 munmap(0x7f2d20543000, 32864) = 0
501 1641 17:21:54.672385 close(4) = 0
502 1641 17:21:54.672661 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
503 1641 17:21:55.174318 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
504 1641 17:21:55.174822 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
505 1641 17:21:55.175314 write(1, "\n", 1) = 1
506 1641 17:21:55.175853 write(1, "[GAME: lol] Processing shot from...", 42) = 42
507 1641 17:21:55.176246 write(1, "[SHOT] Player 0 shoots [1,9] -> "..., 62) = 62
508 1641 17:21:55.176582 write(1, "HIT [1,9]\n", 10) = 10
509 1641 17:21:55.176874 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
510 1641 17:21:55.177204 write(1, "Result sent (version 8)\n", 24) = 24
511 1641 17:21:55.177536 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
512 1641 17:21:55.177833 munmap(0x7f2d20543000, 32864) = 0
513 1641 17:21:55.178327 close(4) = 0
514 1641 17:21:55.178775 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0

```

```

515 1641 17:21:55.697931 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
516 1641 17:21:55.698618 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
517 1641 17:21:55.699163 munmap(0x7f2d20543000, 32864) = 0
518 1641 17:21:55.699452 close(4) = 0
519 1641 17:21:55.699782 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
520 1641 17:21:56.201193 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
521 1641 17:21:56.201865 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
522 1641 17:21:56.202160 munmap(0x7f2d20543000, 32864) = 0
523 1641 17:21:56.202443 close(4) = 0
524 1641 17:21:56.202696 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
525 1641 17:21:56.707992 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
526 1641 17:21:56.708725 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
527 1641 17:21:56.709274 munmap(0x7f2d20543000, 32864) = 0
528 1641 17:21:56.709818 close(4) = 0
529 1641 17:21:56.710189 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
530 1641 17:21:57.218512 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
531 1641 17:21:57.219209 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
532 1641 17:21:57.219890 write(1, "\n", 1) = 1
533 1641 17:21:57.220313 write(1, "[GAME: lol] Processing shot from...", 42) = 42
534 1641 17:21:57.220914 write(1, "[SHOT] Player 0 shoots [2,9] -> "..., 62) = 62
535 1641 17:21:57.221405 write(1, "HIT [2,9]\n", 10) = 10
536 1641 17:21:57.221901 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
537 1641 17:21:57.222265 write(1, "Result sent (version 10)\n", 25) = 25
538 1641 17:21:57.222690 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
539 1641 17:21:57.223041 munmap(0x7f2d20543000, 32864) = 0
540 1641 17:21:57.223451 close(4) = 0
541 1641 17:21:57.223826 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
542 1641 17:21:57.735287 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
543 1641 17:21:57.735883 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
544 1641 17:21:57.736523 munmap(0x7f2d20543000, 32864) = 0
545 1641 17:21:57.737064 close(4) = 0
546 1641 17:21:57.737420 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
547 1641 17:21:58.245745 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
548 1641 17:21:58.246526 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
549 1641 17:21:58.246987 munmap(0x7f2d20543000, 32864) = 0
550 1641 17:21:58.247354 close(4) = 0
551 1641 17:21:58.247728 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
552 1641 17:21:58.754102 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
553 1641 17:21:58.754671 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)

```

```

    = 0x7f2d20543000
554 1641 17:21:58.755105 munmap(0x7f2d20543000, 32864) = 0
555 1641 17:21:58.755474 close(4) = 0
556 1641 17:21:58.755798 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
557 1641 17:21:59.259912 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
558 1641 17:21:59.260432 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
559 1641 17:21:59.260861 munmap(0x7f2d20543000, 32864) = 0
560 1641 17:21:59.261326 close(4) = 0
561 1641 17:21:59.261988 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
562 1641 17:21:59.763441 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
563 1641 17:21:59.763944 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
564 1641 17:21:59.764457 write(1, "\n", 1) = 1
565 1641 17:21:59.764967 write(1, "[GAME: lol] Processing shot from...", 42) = 42
566 1641 17:21:59.765439 write(1, "[SHOT] Player 0 shoots [3,1] -> ", 62) = 62
567 1641 17:21:59.766022 write(1, "HIT [3,1]\n", 10) = 10
568 1641 17:21:59.766446 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
569 1641 17:21:59.766889 write(1, "Result sent (version 12)\n", 25) = 25
570 1641 17:21:59.767354 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
571 1641 17:21:59.767691 munmap(0x7f2d20543000, 32864) = 0
572 1641 17:21:59.768174 close(4) = 0
573 1641 17:21:59.768461 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
574 1641 17:22:00.270054 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
575 1641 17:22:00.270828 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
576 1641 17:22:00.271231 munmap(0x7f2d20543000, 32864) = 0
577 1641 17:22:00.271547 close(4) = 0
578 1641 17:22:00.271886 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
579 1641 17:22:00.773759 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
580 1641 17:22:00.774508 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
581 1641 17:22:00.774887 munmap(0x7f2d20543000, 32864) = 0
582 1641 17:22:00.775220 close(4) = 0
583 1641 17:22:00.775457 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
584 1641 17:22:01.275995 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
585 1641 17:22:01.276358 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
586 1641 17:22:01.276944 munmap(0x7f2d20543000, 32864) = 0
587 1641 17:22:01.277374 close(4) = 0
588 1641 17:22:01.277957 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
589 1641 17:22:01.779164 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
590 1641 17:22:01.779902 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
591 1641 17:22:01.780649 munmap(0x7f2d20543000, 32864) = 0

```

```

592 1641 17:22:01.781091 close(4) = 0
593 1641 17:22:01.781505 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
594 1641 17:22:02.314158 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
595 1641 17:22:02.314749 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
596 1641 17:22:02.315282 write(1, "\n", 1) = 1
597 1641 17:22:02.315757 write(1, "[GAME: lol] Processing shot from"... , 42) = 42
598 1641 17:22:02.316278 write(1, "[SHOT] Player 0 shoots [3,3] -> "... , 63) = 63
599 1641 17:22:02.316704 write(1, "SUNK [3,3]\n", 11) = 11
600 1641 17:22:02.317086 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
601 1641 17:22:02.317504 write(1, "Result sent (version 14)\n", 25) = 25
602 1641 17:22:02.317833 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
603 1641 17:22:02.318249 munmap(0x7f2d20543000, 32864) = 0
604 1641 17:22:02.318625 close(4) = 0
605 1641 17:22:02.318904 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
606 1641 17:22:02.832356 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
607 1641 17:22:02.833037 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
608 1641 17:22:02.833653 munmap(0x7f2d20543000, 32864) = 0
609 1641 17:22:02.834032 close(4) = 0
610 1641 17:22:02.834346 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
611 1641 17:22:03.336099 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
612 1641 17:22:03.336755 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
613 1641 17:22:03.337335 munmap(0x7f2d20543000, 32864) = 0
614 1641 17:22:03.337741 close(4) = 0
615 1641 17:22:03.338095 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
616 1641 17:22:03.854922 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
617 1641 17:22:03.855371 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
618 1641 17:22:03.855761 munmap(0x7f2d20543000, 32864) = 0
619 1641 17:22:03.856143 close(4) = 0
620 1641 17:22:03.856446 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
621 1641 17:22:04.373398 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
622 1641 17:22:04.374279 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
623 1641 17:22:04.375022 munmap(0x7f2d20543000, 32864) = 0
624 1641 17:22:04.375388 close(4) = 0
625 1641 17:22:04.375691 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
626 1641 17:22:04.891062 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
627 1641 17:22:04.891843 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
628 1641 17:22:04.892561 munmap(0x7f2d20543000, 32864) = 0
629 1641 17:22:04.893075 close(4) = 0
630 1641 17:22:04.893534 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,

```

```

        tv_nsec=500000000}, NULL) = 0
631 1641 17:22:05.395678 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
632 1641 17:22:05.396396 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
633 1641 17:22:05.396921 write(1, "\n", 1) = 1
634 1641 17:22:05.397533 write(1, "[GAME: lol] Processing shot from...", 42) = 42
635 1641 17:22:05.398016 write(1, "[SHOT] Player 0 shoots [3,9] -> "..., 63) = 63
636 1641 17:22:05.398465 write(1, "SUNK [3,9]\n", 11) = 11
637 1641 17:22:05.398878 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
638 1641 17:22:05.399313 write(1, "Result sent (version 16)\n", 25) = 25
639 1641 17:22:05.399705 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
640 1641 17:22:05.400087 munmap(0x7f2d20543000, 32864) = 0
641 1641 17:22:05.400395 close(4) = 0
642 1641 17:22:05.400697 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
643 1641 17:22:05.921379 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
644 1641 17:22:05.921921 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
645 1641 17:22:05.922419 munmap(0x7f2d20543000, 32864) = 0
646 1641 17:22:05.922760 close(4) = 0
647 1641 17:22:05.923159 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
648 1641 17:22:06.431929 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
649 1641 17:22:06.432708 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
650 1641 17:22:06.433206 munmap(0x7f2d20543000, 32864) = 0
651 1641 17:22:06.433575 close(4) = 0
652 1641 17:22:06.433928 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
653 1641 17:22:06.935781 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
654 1641 17:22:06.936484 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
655 1641 17:22:06.936956 munmap(0x7f2d20543000, 32864) = 0
656 1641 17:22:06.937443 close(4) = 0
657 1641 17:22:06.937848 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
658 1641 17:22:07.439175 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
659 1641 17:22:07.439621 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
660 1641 17:22:07.440095 munmap(0x7f2d20543000, 32864) = 0
661 1641 17:22:07.440505 close(4) = 0
662 1641 17:22:07.440873 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
663 1641 17:22:07.964616 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
664 1641 17:22:07.965266 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
665 1641 17:22:07.965726 munmap(0x7f2d20543000, 32864) = 0
666 1641 17:22:07.966053 close(4) = 0
667 1641 17:22:07.966371 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
668 1641 17:22:08.506007 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4

```

```

669 1641 17:22:08.506818 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
670 1641 17:22:08.507496 write(1, "\n", 1) = 1
671 1641 17:22:08.507900 write(1, "[GAME: lol] Processing shot from...", 42) = 42
672 1641 17:22:08.508300 write(1, "[SHOT] Player 0 shoots [4,1] -> "..., 63) = 63
673 1641 17:22:08.508757 write(1, "SUNK [4,1]\n", 11) = 11
674 1641 17:22:08.509295 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
675 1641 17:22:08.509640 write(1, "Result sent (version 18)\n", 25) = 25
676 1641 17:22:08.510063 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
677 1641 17:22:08.510375 munmap(0x7f2d20543000, 32864) = 0
678 1641 17:22:08.510835 close(4) = 0
679 1641 17:22:08.511240 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
680 1641 17:22:09.012818 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
681 1641 17:22:09.013579 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
682 1641 17:22:09.014098 munmap(0x7f2d20543000, 32864) = 0
683 1641 17:22:09.014433 close(4) = 0
684 1641 17:22:09.014700 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
685 1641 17:22:09.516110 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
686 1641 17:22:09.516777 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
687 1641 17:22:09.517181 munmap(0x7f2d20543000, 32864) = 0
688 1641 17:22:09.517525 close(4) = 0
689 1641 17:22:09.517914 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
690 1641 17:22:10.019115 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
691 1641 17:22:10.019757 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
692 1641 17:22:10.020326 munmap(0x7f2d20543000, 32864) = 0
693 1641 17:22:10.020799 close(4) = 0
694 1641 17:22:10.021241 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
695 1641 17:22:10.528841 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
696 1641 17:22:10.529551 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
697 1641 17:22:10.530137 munmap(0x7f2d20543000, 32864) = 0
698 1641 17:22:10.530573 close(4) = 0
699 1641 17:22:10.530867 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
700 1641 17:22:11.032592 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
701 1641 17:22:11.033298 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
702 1641 17:22:11.033810 munmap(0x7f2d20543000, 32864) = 0
703 1641 17:22:11.034283 close(4) = 0
704 1641 17:22:11.034728 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
705 1641 17:22:11.535369 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
706 1641 17:22:11.535907 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000

```

```

707 1641 17:22:11.536411 write(1, "\n", 1) = 1
708 1641 17:22:11.536688 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
709 1641 17:22:11.537085 write(1, "[SHOT] Player 0 shoots [5,6] -> "..., 62) = 62
710 1641 17:22:11.537450 write(1, "HIT [5,6]\n", 10) = 10
711 1641 17:22:11.537727 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
712 1641 17:22:11.538040 write(1, "Result sent (version 20)\n", 25) = 25
713 1641 17:22:11.538370 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
714 1641 17:22:11.538638 munmap(0x7f2d20543000, 32864) = 0
715 1641 17:22:11.538955 close(4) = 0
716 1641 17:22:11.539189 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
717 1641 17:22:12.042297 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
718 1641 17:22:12.042972 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
719 1641 17:22:12.043448 munmap(0x7f2d20543000, 32864) = 0
720 1641 17:22:12.043851 close(4) = 0
721 1641 17:22:12.044219 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
722 1641 17:22:12.554604 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
723 1641 17:22:12.555232 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
724 1641 17:22:12.555821 munmap(0x7f2d20543000, 32864) = 0
725 1641 17:22:12.556297 close(4) = 0
726 1641 17:22:12.556689 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
727 1641 17:22:13.057947 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
728 1641 17:22:13.058819 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
729 1641 17:22:13.059201 munmap(0x7f2d20543000, 32864) = 0
730 1641 17:22:13.059649 close(4) = 0
731 1641 17:22:13.060109 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
732 1641 17:22:13.562870 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
733 1641 17:22:13.563354 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
734 1641 17:22:13.563878 write(1, "\n", 1) = 1
735 1641 17:22:13.564439 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
736 1641 17:22:13.564964 write(1, "[SHOT] Player 0 shoots [5,7] -> "..., 63) = 63
737 1641 17:22:13.565407 write(1, "SUNK [5,7]\n", 11) = 11
738 1641 17:22:13.565749 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
739 1641 17:22:13.566075 write(1, "Result sent (version 22)\n", 25) = 25
740 1641 17:22:13.566347 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
741 1641 17:22:13.566645 munmap(0x7f2d20543000, 32864) = 0
742 1641 17:22:13.566890 close(4) = 0
743 1641 17:22:13.567132 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
744 1641 17:22:14.072765 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
745 1641 17:22:14.073252 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
746 1641 17:22:14.073644 munmap(0x7f2d20543000, 32864) = 0
747 1641 17:22:14.073999 close(4) = 0

```

```
748 1641 17:22:14.074361 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
749 1641 17:22:14.575431 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
750 1641 17:22:14.575958 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
751 1641 17:22:14.576466 munmap(0x7f2d20543000, 32864) = 0
752 1641 17:22:14.576902 close(4) = 0
753 1641 17:22:14.577233 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
754 1641 17:22:15.101205 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
755 1641 17:22:15.101813 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
756 1641 17:22:15.102321 munmap(0x7f2d20543000, 32864) = 0
757 1641 17:22:15.102717 close(4) = 0
758 1641 17:22:15.103053 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
759 1641 17:22:15.604815 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
760 1641 17:22:15.605337 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
761 1641 17:22:15.605817 munmap(0x7f2d20543000, 32864) = 0
762 1641 17:22:15.606184 close(4) = 0
763 1641 17:22:15.606601 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
764 1641 17:22:16.109853 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
765 1641 17:22:16.110523 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
766 1641 17:22:16.111225 munmap(0x7f2d20543000, 32864) = 0
767 1641 17:22:16.111657 close(4) = 0
768 1641 17:22:16.112025 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
769 1641 17:22:16.613242 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
770 1641 17:22:16.613824 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
771 1641 17:22:16.614441 munmap(0x7f2d20543000, 32864) = 0
772 1641 17:22:16.614721 close(4) = 0
773 1641 17:22:16.615101 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
774 1641 17:22:17.140766 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
775 1641 17:22:17.141313 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
776 1641 17:22:17.141629 munmap(0x7f2d20543000, 32864) = 0
777 1641 17:22:17.142014 close(4) = 0
778 1641 17:22:17.142371 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
779 1641 17:22:17.643638 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
780 1641 17:22:17.644196 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
781 1641 17:22:17.644556 munmap(0x7f2d20543000, 32864) = 0
782 1641 17:22:17.644974 close(4) = 0
783 1641 17:22:17.645344 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
```

```
784 1641 17:22:18.148945 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
785 1641 17:22:18.149586 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
786 1641 17:22:18.150031 write(1, "\n", 1) = 1
787 1641 17:22:18.150343 write(1, "[GAME: lol] Processing shot from...", 42) = 42
788 1641 17:22:18.150818 write(1, "[SHOT] Player 0 shoots [6,1] -> ", 62) = 62
789 1641 17:22:18.151306 write(1, "HIT [6,1]\n", 10) = 10
790 1641 17:22:18.151832 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
791 1641 17:22:18.152230 write(1, "Result sent (version 24)\n", 25) = 25
792 1641 17:22:18.152624 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
793 1641 17:22:18.152991 munmap(0x7f2d20543000, 32864) = 0
794 1641 17:22:18.153283 close(4) = 0
795 1641 17:22:18.153516 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
796 1641 17:22:18.654606 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
797 1641 17:22:18.654995 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
798 1641 17:22:18.655476 munmap(0x7f2d20543000, 32864) = 0
799 1641 17:22:18.655888 close(4) = 0
800 1641 17:22:18.656369 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
801 1641 17:22:17.343723 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
802 1641 17:22:17.344293 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
803 1641 17:22:17.344631 munmap(0x7f2d20543000, 32864) = 0
804 1641 17:22:17.344980 close(4) = 0
805 1641 17:22:17.345488 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
806 1641 17:22:17.859690 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
807 1641 17:22:17.860464 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
808 1641 17:22:17.861019 munmap(0x7f2d20543000, 32864) = 0
809 1641 17:22:17.861387 close(4) = 0
810 1641 17:22:17.861858 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
811 1641 17:22:18.362544 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
812 1641 17:22:18.363134 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
813 1641 17:22:18.363685 munmap(0x7f2d20543000, 32864) = 0
814 1641 17:22:18.364017 close(4) = 0
815 1641 17:22:18.364433 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
816 1641 17:22:18.875469 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
817 1641 17:22:18.875947 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
818 1641 17:22:18.876255 munmap(0x7f2d20543000, 32864) = 0
819 1641 17:22:18.876565 close(4) = 0
820 1641 17:22:18.876857 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
821 1641 17:22:19.378024 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
822 1641 17:22:19.378782 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```

    = 0x7f2d20543000
823 1641 17:22:19.379365 munmap(0x7f2d20543000, 32864) = 0
824 1641 17:22:19.379769 close(4) = 0
825 1641 17:22:19.380133 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
826 1641 17:22:19.883744 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
827 1641 17:22:19.884406 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
828 1641 17:22:19.884914 munmap(0x7f2d20543000, 32864) = 0
829 1641 17:22:19.885351 close(4) = 0
830 1641 17:22:19.885825 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
831 1641 17:22:20.387301 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
832 1641 17:22:20.388039 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
833 1641 17:22:20.388639 munmap(0x7f2d20543000, 32864) = 0
834 1641 17:22:20.389044 close(4) = 0
835 1641 17:22:20.389427 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
836 1641 17:22:20.892165 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
837 1641 17:22:20.892738 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
838 1641 17:22:20.893278 munmap(0x7f2d20543000, 32864) = 0
839 1641 17:22:20.893676 close(4) = 0
840 1641 17:22:20.894001 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
841 1641 17:22:21.395579 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
842 1641 17:22:21.396380 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
843 1641 17:22:21.396864 write(1, "\n", 1) = 1
844 1641 17:22:21.397314 write(1, "[GAME: lol] Processing shot from...", 42) = 42
845 1641 17:22:21.397697 write(1, "[SHOT] Player 0 shoots [6,2] -> "..., 62) = 62
846 1641 17:22:21.398092 write(1, "HIT [6,2]\n", 10) = 10
847 1641 17:22:21.398479 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
848 1641 17:22:21.399011 write(1, "Result sent (version 26)\n", 25) = 25
849 1641 17:22:21.399484 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
850 1641 17:22:21.399838 munmap(0x7f2d20543000, 32864) = 0
851 1641 17:22:21.400115 close(4) = 0
852 1641 17:22:21.400399 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
853 1641 17:22:21.904610 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
854 1641 17:22:21.905880 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
855 1641 17:22:21.906282 munmap(0x7f2d20543000, 32864) = 0
856 1641 17:22:21.906574 close(4) = 0
857 1641 17:22:21.906903 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
858 1641 17:22:22.408063 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
859 1641 17:22:22.408590 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
860 1641 17:22:22.409019 munmap(0x7f2d20543000, 32864) = 0

```

```

861 1641 17:22:22.409304 close(4) = 0
862 1641 17:22:22.409572 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
863 1641 17:22:22.917533 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
864 1641 17:22:22.918192 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
865 1641 17:22:22.918629 munmap(0x7f2d20543000, 32864) = 0
866 1641 17:22:22.919044 close(4) = 0
867 1641 17:22:22.919439 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
868 1641 17:22:23.421324 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
869 1641 17:22:23.421969 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
870 1641 17:22:23.422480 write(1, "\n", 1) = 1
871 1641 17:22:23.422885 write(1, "[GAME: lol] Processing shot from...", 42) = 42
872 1641 17:22:23.423172 write(1, "[SHOT] Player 0 shoots [6,3] -> "..., 63) = 63
873 1641 17:22:23.423560 write(1, "SUNK [6,3]\n", 11) = 11
874 1641 17:22:23.423963 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
875 1641 17:22:23.424531 write(1, "Result sent (version 28)\n", 25) = 25
876 1641 17:22:23.424967 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
877 1641 17:22:23.425382 munmap(0x7f2d20543000, 32864) = 0
878 1641 17:22:23.425795 close(4) = 0
879 1641 17:22:23.426202 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
880 1641 17:22:23.928629 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
881 1641 17:22:23.929318 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
882 1641 17:22:23.929903 munmap(0x7f2d20543000, 32864) = 0
883 1641 17:22:23.930367 close(4) = 0
884 1641 17:22:23.930769 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
885 1641 17:22:24.432742 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
886 1641 17:22:24.433232 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
887 1641 17:22:24.433831 munmap(0x7f2d20543000, 32864) = 0
888 1641 17:22:24.434232 close(4) = 0
889 1641 17:22:24.434696 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
890 1641 17:22:24.940698 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
891 1641 17:22:24.941239 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
892 1641 17:22:24.941857 munmap(0x7f2d20543000, 32864) = 0
893 1641 17:22:24.942201 close(4) = 0
894 1641 17:22:24.942495 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
895 1641 17:22:25.443971 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
896 1641 17:22:25.444890 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
897 1641 17:22:25.445289 munmap(0x7f2d20543000, 32864) = 0
898 1641 17:22:25.445600 close(4) = 0
899 1641 17:22:25.446014 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,

```

```

    tv_nsec=500000000}, NULL) = 0
900 1641 17:22:25.955721 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
901 1641 17:22:25.956482 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
902 1641 17:22:25.956941 munmap(0x7f2d20543000, 32864) = 0
903 1641 17:22:25.957231 close(4) = 0
904 1641 17:22:25.957542 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
905 1641 17:22:26.466486 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
906 1641 17:22:26.466918 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
907 1641 17:22:26.467466 write(1, "\n", 1) = 1
908 1641 17:22:26.467940 write(1, "[GAME: lol] Processing shot from...", 42) = 42
909 1641 17:22:26.468378 write(1, "[SHOT] Player 0 shoots [7,8] -> ", 63) = 63
910 1641 17:22:26.468816 write(1, "SUNK [7,8]\n", 11) = 11
911 1641 17:22:26.469186 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
912 1641 17:22:26.469610 write(1, "Result sent (version 30)\n", 25) = 25
913 1641 17:22:26.470033 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
914 1641 17:22:26.470347 munmap(0x7f2d20543000, 32864) = 0
915 1641 17:22:26.470651 close(4) = 0
916 1641 17:22:26.470964 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
917 1641 17:22:26.979726 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
918 1641 17:22:26.980367 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
919 1641 17:22:26.980995 munmap(0x7f2d20543000, 32864) = 0
920 1641 17:22:26.981279 close(4) = 0
921 1641 17:22:26.981555 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
922 1641 17:22:27.484309 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
923 1641 17:22:27.484782 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
924 1641 17:22:27.485210 munmap(0x7f2d20543000, 32864) = 0
925 1641 17:22:27.485618 close(4) = 0
926 1641 17:22:27.485963 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
927 1641 17:22:28.006492 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
928 1641 17:22:28.007209 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
929 1641 17:22:28.007862 munmap(0x7f2d20543000, 32864) = 0
930 1641 17:22:28.008305 close(4) = 0
931 1641 17:22:28.008727 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
932 1641 17:22:28.510573 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
933 1641 17:22:28.511512 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
934 1641 17:22:28.511975 munmap(0x7f2d20543000, 32864) = 0
935 1641 17:22:28.512390 close(4) = 0
936 1641 17:22:28.512671 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
937 1641 17:22:29.016456 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4

```

```
938 1641 17:22:29.017130 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
939 1641 17:22:29.017597 munmap(0x7f2d20543000, 32864) = 0
940 1641 17:22:29.018092 close(4) = 0
941 1641 17:22:29.018391 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
942 1641 17:22:29.523191 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
943 1641 17:22:29.524092 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
944 1641 17:22:29.524512 write(1, "\n", 1) = 1
945 1641 17:22:29.524875 write(1, "[GAME: lol] Processing shot from...", 42) = 42
946 1641 17:22:29.525384 write(1, "[SHOT] Player 0 shoots [8,0] -> "..., 63) = 63
947 1641 17:22:29.525817 write(1, "SUNK [8,0]\n", 11) = 11
948 1641 17:22:29.526192 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
949 1641 17:22:29.526585 write(1, "Result sent (version 32)\n", 25) = 25
950 1641 17:22:29.526940 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
951 1641 17:22:29.527216 munmap(0x7f2d20543000, 32864) = 0
952 1641 17:22:29.527529 close(4) = 0
953 1641 17:22:29.527862 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
954 1641 17:22:30.043227 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
955 1641 17:22:30.043969 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
956 1641 17:22:30.044454 munmap(0x7f2d20543000, 32864) = 0
957 1641 17:22:30.044767 close(4) = 0
958 1641 17:22:30.045037 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
959 1641 17:22:30.549731 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
960 1641 17:22:30.550583 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
961 1641 17:22:30.551185 munmap(0x7f2d20543000, 32864) = 0
962 1641 17:22:30.551462 close(4) = 0
963 1641 17:22:30.551842 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
964 1641 17:22:31.058657 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
965 1641 17:22:31.059161 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
966 1641 17:22:31.059510 munmap(0x7f2d20543000, 32864) = 0
967 1641 17:22:31.059909 close(4) = 0
968 1641 17:22:31.060298 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
969 1641 17:22:31.591525 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
970 1641 17:22:31.592164 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
971 1641 17:22:31.592634 munmap(0x7f2d20543000, 32864) = 0
972 1641 17:22:31.593133 close(4) = 0
973 1641 17:22:31.593722 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
974 1641 17:22:32.106783 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
975 1641 17:22:32.107481 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
```

```
976 1641 17:22:32.107916 munmap(0x7f2d20543000, 32864) = 0
977 1641 17:22:32.108202 close(4) = 0
978 1641 17:22:32.108512 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
979 1641 17:22:32.610190 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
980 1641 17:22:32.611003 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
981 1641 17:22:32.611620 munmap(0x7f2d20543000, 32864) = 0
982 1641 17:22:32.612034 close(4) = 0
983 1641 17:22:32.612410 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
984 1641 17:22:33.123710 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
985 1641 17:22:33.124403 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
986 1641 17:22:33.124845 munmap(0x7f2d20543000, 32864) = 0
987 1641 17:22:33.125155 close(4) = 0
988 1641 17:22:33.125451 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
989 1641 17:22:33.637008 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
990 1641 17:22:33.637698 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
991 1641 17:22:33.638280 munmap(0x7f2d20543000, 32864) = 0
992 1641 17:22:33.638641 close(4) = 0
993 1641 17:22:33.638946 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
994 1641 17:22:34.147241 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
995 1641 17:22:34.147940 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
996 1641 17:22:34.148326 munmap(0x7f2d20543000, 32864) = 0
997 1641 17:22:34.148628 close(4) = 0
998 1641 17:22:34.148923 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
999 1641 17:22:34.650132 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1000 1641 17:22:34.650474 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1001 1641 17:22:34.650930 munmap(0x7f2d20543000, 32864) = 0
1002 1641 17:22:34.651289 close(4) = 0
1003 1641 17:22:34.651635 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1004 1641 17:22:35.164552 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1005 1641 17:22:35.165240 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1006 1641 17:22:35.165691 write(1, "\n", 1) = 1
1007 1641 17:22:35.166105 write(1, "[GAME: lol] Processing shot from...", 42) = 42
1008 1641 17:22:35.166475 write(1, "[SHOT] Player 0 shoots [9,5] -> ", 62) = 62
1009 1641 17:22:35.166846 write(1, "HIT [9,5]\n", 10) = 10
1010 1641 17:22:35.167296 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1011 1641 17:22:35.167863 write(1, "Result sent (version 34)\n", 25) = 25
1012 1641 17:22:35.168208 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1013 1641 17:22:35.168487 munmap(0x7f2d20543000, 32864) = 0
1014 1641 17:22:35.168799 close(4) = 0
```

```
1015 1641 17:22:35.169143 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1016 1641 17:22:35.689625 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1017 1641 17:22:35.690428 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1018 1641 17:22:35.691132 munmap(0x7f2d20543000, 32864) = 0
1019 1641 17:22:35.691630 close(4) = 0
1020 1641 17:22:35.691955 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1021 1641 17:22:36.193098 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1022 1641 17:22:36.193898 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1023 1641 17:22:36.194433 munmap(0x7f2d20543000, 32864) = 0
1024 1641 17:22:36.194719 close(4) = 0
1025 1641 17:22:36.194991 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1026 1641 17:22:36.696333 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1027 1641 17:22:36.696988 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1028 1641 17:22:36.697318 munmap(0x7f2d20543000, 32864) = 0
1029 1641 17:22:36.697835 close(4) = 0
1030 1641 17:22:36.698259 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1031 1641 17:22:37.214641 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1032 1641 17:22:37.215462 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1033 1641 17:22:37.215913 munmap(0x7f2d20543000, 32864) = 0
1034 1641 17:22:37.216239 close(4) = 0
1035 1641 17:22:37.216697 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1036 1641 17:22:37.748095 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1037 1641 17:22:37.748769 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1038 1641 17:22:37.749289 munmap(0x7f2d20543000, 32864) = 0
1039 1641 17:22:37.749727 close(4) = 0
1040 1641 17:22:37.750160 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1041 1641 17:22:38.250836 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1042 1641 17:22:38.251151 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1043 1641 17:22:38.251436 munmap(0x7f2d20543000, 32864) = 0
1044 1641 17:22:38.251692 close(4) = 0
1045 1641 17:22:38.251948 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1046 1641 17:22:38.752598 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1047 1641 17:22:38.752900 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1048 1641 17:22:38.753110 write(1, "\n", 1) = 1
1049 1641 17:22:38.753436 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1050 1641 17:22:38.753744 write(1, "[SHOT] Player 0 shoots [9,4] -> "..., 62) = 62
1051 1641 17:22:38.754126 write(1, "HIT [9,4]\n", 10) = 10
```

```
1052 1641 17:22:38.754410 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1053 1641 17:22:38.754684 write(1, "Result sent (version 36)\n", 25) = 25
1054 1641 17:22:38.754877 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1055 1641 17:22:38.755099 munmap(0x7f2d20543000, 32864) = 0
1056 1641 17:22:38.755292 close(4) = 0
1057 1641 17:22:38.755485 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1058 1641 17:22:39.273922 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1059 1641 17:22:39.274615 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1060 1641 17:22:39.275097 munmap(0x7f2d20543000, 32864) = 0
1061 1641 17:22:39.275416 close(4) = 0
1062 1641 17:22:39.275740 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1063 1641 17:22:39.798908 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1064 1641 17:22:39.799582 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1065 1641 17:22:39.800152 munmap(0x7f2d20543000, 32864) = 0
1066 1641 17:22:39.800460 close(4) = 0
1067 1641 17:22:39.800810 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1068 1641 17:22:40.303603 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1069 1641 17:22:40.303959 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1070 1641 17:22:40.304459 munmap(0x7f2d20543000, 32864) = 0
1071 1641 17:22:40.305016 close(4) = 0
1072 1641 17:22:40.305411 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1073 1641 17:22:40.806372 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1074 1641 17:22:40.806665 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1075 1641 17:22:40.807109 write(1, "\n", 1) = 1
1076 1641 17:22:40.807532 write(1, "[GAME: lol] Processing shot from...", 42) = 42
1077 1641 17:22:40.808037 write(1, "[SHOT] Player 0 shoots [9,6] -> "..., 62) = 62
1078 1641 17:22:40.808358 write(1, "HIT [9,6]\n", 10) = 10
1079 1641 17:22:40.808771 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1080 1641 17:22:40.809147 write(1, "Result sent (version 38)\n", 25) = 25
1081 1641 17:22:40.809544 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1082 1641 17:22:40.809845 munmap(0x7f2d20543000, 32864) = 0
1083 1641 17:22:40.810023 close(4) = 0
1084 1641 17:22:40.810214 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1085 1641 17:22:41.320877 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1086 1641 17:22:41.321114 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1087 1641 17:22:41.321551 munmap(0x7f2d20543000, 32864) = 0
1088 1641 17:22:41.321815 close(4) = 0
1089 1641 17:22:41.321985 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1090 1641 17:22:41.827772 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1091 1641 17:22:41.828261 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

[illegible]

```
1126 1641 17:22:42.847699 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1127 1641 17:22:43.349044 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1128 1641 17:22:43.349809 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1129 1641 17:22:43.350518 munmap(0x7f2d20543000, 32864) = 0
1130 1641 17:22:43.350945 close(4) = 0
1131 1641 17:22:43.351267 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1132 1641 17:22:43.856727 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1133 1641 17:22:43.857146 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1134 1641 17:22:43.857667 munmap(0x7f2d20543000, 32864) = 0
1135 1641 17:22:43.858164 close(4) = 0
1136 1641 17:22:43.858553 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1137 1641 17:22:44.359417 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1138 1641 17:22:44.359794 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1139 1641 17:22:44.360098 munmap(0x7f2d20543000, 32864) = 0
1140 1641 17:22:44.360416 close(4) = 0
1141 1641 17:22:44.360851 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1142 1641 17:22:44.867261 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1143 1641 17:22:44.867749 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1144 1641 17:22:44.868309 munmap(0x7f2d20543000, 32864) = 0
1145 1641 17:22:44.868776 close(4) = 0
1146 1641 17:22:44.869218 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1147 1641 17:22:45.374290 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1148 1641 17:22:45.375023 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1149 1641 17:22:45.375531 munmap(0x7f2d20543000, 32864) = 0
1150 1641 17:22:45.375990 close(4) = 0
1151 1641 17:22:45.376386 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1152 1641 17:22:45.885522 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1153 1641 17:22:45.886405 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1154 1641 17:22:45.887186 munmap(0x7f2d20543000, 32864) = 0
1155 1641 17:22:45.887720 close(4) = 0
1156 1641 17:22:45.888170 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1157 1641 17:22:46.392853 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1158 1641 17:22:46.393605 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1159 1641 17:22:46.394159 munmap(0x7f2d20543000, 32864) = 0
1160 1641 17:22:46.394556 close(4) = 0
1161 1641 17:22:46.394943 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
```

```
1162 1641 17:22:46.904342 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1163 1641 17:22:46.905038 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1164 1641 17:22:46.905495 munmap(0x7f2d20543000, 32864) = 0
1165 1641 17:22:46.905941 close(4) = 0
1166 1641 17:22:46.906276 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1167 1641 17:22:47.409673 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1168 1641 17:22:47.410082 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1169 1641 17:22:47.410475 munmap(0x7f2d20543000, 32864) = 0
1170 1641 17:22:47.410757 close(4) = 0
1171 1641 17:22:47.411010 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1172 1641 17:22:47.920557 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1173 1641 17:22:47.921343 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1174 1641 17:22:47.922061 munmap(0x7f2d20543000, 32864) = 0
1175 1641 17:22:47.922720 close(4) = 0
1176 1641 17:22:47.923048 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1177 1641 17:22:48.424989 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1178 1641 17:22:48.425303 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1179 1641 17:22:48.425612 munmap(0x7f2d20543000, 32864) = 0
1180 1641 17:22:48.425950 close(4) = 0
1181 1641 17:22:48.426301 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1182 1641 17:22:48.930935 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1183 1641 17:22:48.931614 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1184 1641 17:22:48.932272 munmap(0x7f2d20543000, 32864) = 0
1185 1641 17:22:48.932690 close(4) = 0
1186 1641 17:22:48.933077 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1187 1641 17:22:49.434374 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1188 1641 17:22:49.434622 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1189 1641 17:22:49.435018 munmap(0x7f2d20543000, 32864) = 0
1190 1641 17:22:49.435619 close(4) = 0
1191 1641 17:22:49.436046 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1192 1641 17:22:48.174989 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1193 1641 17:22:48.175649 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
      = 0x7f2d20543000
1194 1641 17:22:48.176069 munmap(0x7f2d20543000, 32864) = 0
1195 1641 17:22:48.176439 close(4) = 0
1196 1641 17:22:48.176757 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
      tv_nsec=500000000}, NULL) = 0
1197 1641 17:22:48.679050 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1198 1641 17:22:48.679308 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```

= 0x7f2d20543000
1199 1641 17:22:48.679563 munmap(0x7f2d20543000, 32864) = 0
1200 1641 17:22:48.679855 close(4) = 0
1201 1641 17:22:48.680079 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=5000000000}, NULL) = 0
1202 1641 17:22:49.184478 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1203 1641 17:22:49.185221 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1204 1641 17:22:49.185594 munmap(0x7f2d20543000, 32864) = 0
1205 1641 17:22:49.185981 close(4) = 0
1206 1641 17:22:49.186355 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=5000000000}, NULL) = 0
1207 1641 17:22:49.692015 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1208 1641 17:22:49.692315 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1209 1641 17:22:49.692713 munmap(0x7f2d20543000, 32864) = 0
1210 1641 17:22:49.693009 close(4) = 0
1211 1641 17:22:49.693384 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=5000000000}, NULL) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1212 1641 17:22:50.194038 --- SIGINT {si_signo=SIGINT, si_code=SI_KERNEL} ---
1213 1641 17:22:50.194260 rt_sigreturn({mask=[]}) = -1 EINTR (Interrupted system
    call)
1214 1641 17:22:50.194727 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1215 1641 17:22:50.195336 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1216 1641 17:22:50.195823 munmap(0x7f2d20543000, 32864) = 0
1217 1641 17:22:50.196185 close(4) = 0
1218 1641 17:22:50.196664 write(1, "\n", 1) = 1
1219 1641 17:22:50.197153 write(1, "[SERVER] Shutting down...\n", 26) = 26
1220 1641 17:22:50.197672 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=5, tv_nsec=0},
    NULL) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1221 1641 17:22:53.077456 --- SIGINT {si_signo=SIGINT, si_code=SI_KERNEL} ---
1222 1641 17:22:53.077646 rt_sigreturn({mask=[]}) = -1 EINTR (Interrupted system
    call)
1223 1641 17:22:53.078158 write(1, "Synchronization primitives clean...", 38) = 38
1224 1641 17:22:53.078521 munmap(0x2, 32864) = -1 EINVAL (Invalid argument)
1225 1641 17:22:53.078848 close(395049983) = -1 EBADF (Bad file descriptor)
1226 1641 17:22:53.079210 write(1, "Server stopped\n", 15) = 15
1227 1641 17:22:53.079740 exit_group(0) = ?
1228 1641 17:22:53.080221 +++ exited with 0 +++

```

Листинг 15: *Strace логи сервера*

```

1 1674 17:21:16.783887 execve("./battleship_client", ["./battleship_client"],
    0x7ffc852f56a0 /* 36 vars */) = 0
2 1674 17:21:16.784972 brk(NULL) = 0x558c9ac97000
3 1674 17:21:16.785363 arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe13e80ee0) = -1
    EINVAL (Invalid argument)
4 1674 17:21:16.785747 mmap(NULL, 8192, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f7fa8cd9000

```

```

5 1674 17:21:16.786122 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such
    file or directory)
6 1674 17:21:16.786477 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC)
    = 3
7 1674 17:21:16.787041 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=30404,
    ...}, AT_EMPTY_PATH) = 0
8 1674 17:21:16.787504 mmap(NULL, 30404, PROT_READ, MAP_PRIVATE, 3, 0) =
    0x7f7fa8cd1000
9 1674 17:21:16.787882 close(3) = 0
10 1674 17:21:16.788134 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6",
    O_RDONLY|O_CLOEXEC) = 3
11 1674 17:21:16.788405 read(3,
    "\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"..., 832)
    = 832
12 1674 17:21:16.788646 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"..., 784,
    64) = 784
13 1674 17:21:16.788918 pread64(3, "\4\0\0\0 \
    \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0"..., 48, 848) = 48
14 1674 17:21:16.789225 pread64(3,
    "\4\0\0\0\24\0\0\0\3\0\0\0GNU\00{\f\225\\=\201\327\312\301P\32$\230\266\235"...,
    68, 896) = 68
15 1674 17:21:16.789524 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400,
    ...}, AT_EMPTY_PATH) = 0
16 1674 17:21:16.789800 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"..., 784,
    64) = 784
17 1674 17:21:16.790005 mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE,
    3, 0) = 0x7f7fa8aa8000
18 1674 17:21:16.790306 mprotect(0x7f7fa8ad0000, 2023424, PROT_NONE) = 0
19 1674 17:21:16.790494 mmap(0x7f7fa8ad0000, 1658880, PROT_READ|PROT_EXEC,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f7fa8ad0000
20 1674 17:21:16.790680 mmap(0x7f7fa8c65000, 360448, PROT_READ,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f7fa8c65000
21 1674 17:21:16.790861 mmap(0x7f7fa8cbe000, 24576, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f7fa8cbe000
22 1674 17:21:16.791057 mmap(0x7f7fa8cc4000, 52816, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f7fa8cc4000
23 1674 17:21:16.791278 close(3) = 0
24 1674 17:21:16.791775 mmap(NULL, 12288, PROT_READ|PROT_WRITE,
    MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f7fa8aa5000
25 1674 17:21:16.792093 arch_prctl(ARCH_SET_FS, 0x7f7fa8aa5740) = 0
26 1674 17:21:16.792364 set_tid_address(0x7f7fa8aa5a10) = 1674
27 1674 17:21:16.792783 set_robust_list(0x7f7fa8aa5a20, 24) = 0
28 1674 17:21:16.792999 rseq(0x7f7fa8aa60e0, 0x20, 0, 0x53053053) = 0
29 1674 17:21:16.793770 mprotect(0x7f7fa8cbe000, 16384, PROT_READ) = 0
30 1674 17:21:16.794205 mprotect(0x558c7d3a5000, 4096, PROT_READ) = 0
31 1674 17:21:16.794389 mprotect(0x7f7fa8d13000, 8192, PROT_READ) = 0
32 1674 17:21:16.794548 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
    rlim_max=RLIM64_INFINITY}) = 0
33 1674 17:21:16.794709 munmap(0x7f7fa8cd1000, 30404) = 0

```

[illegible]

```

75 1674 17:21:29.233419 write(1, "Generating random ship placement"... , 36) = 36
76 1674 17:21:29.233828 write(1, "- 1 four-deck ship (1x4)\n", 25) = 25
77 1674 17:21:29.234298 write(1, "- 2 three-deck ships (2x3)\n", 27) = 27
78 1674 17:21:29.234813 write(1, "- 3 two-deck ships (3x2)\n", 25) = 25
79 1674 17:21:29.235273 write(1, "- 4 one-deck ships (4x1)\n", 25) = 25
80 1674 17:21:29.235779 write(1, "\n", 1) = 1
81 1674 17:21:29.236211 write(1, "\n", 1) = 1
82 1674 17:21:29.236668 write(1, "Ships placed randomly!\n", 23) = 23
83 1674 17:21:29.237129 write(1, " 0 1 2 3 4 5 6 7 8 9\n", 23) = 23
84 1674 17:21:29.237491 write(1, " 0 . S . . . . . S S . \n", 24) = 24
85 1674 17:21:29.237870 write(1, " 1 . . . . . . . . . \n", 24) = 24
86 1674 17:21:29.238400 write(1, " 2 S . . . . . S S . \n", 24) = 24
87 1674 17:21:29.238784 write(1, " 3 . . . . . . . . . \n", 24) = 24
88 1674 17:21:29.239105 write(1, " 4 S S S S . . . . . \n", 24) = 24
89 1674 17:21:29.239479 write(1, " 5 . . . . . . S S . \n", 24) = 24
90 1674 17:21:29.239808 write(1, " 6 . S . S . . . . . \n", 24) = 24
91 1674 17:21:29.240186 write(1, " 7 . . . . . . . . . \n", 24) = 24
92 1674 17:21:29.240619 write(1, " 8 S S S . . . S S S . \n", 24) = 24
93 1674 17:21:29.240963 write(1, " 9 . . . . . . . . . \n", 24) = 24
94 1674 17:21:29.241336 msync(0xf7fa8a9c000, 2048, MS_SYNC) = 0
95 1674 17:21:29.243836 write(1, "\n", 1) = 1
96 1674 17:21:29.244256 write(1, "Waiting for second player...\n", 29) = 29
97 1674 17:21:29.244710 write(1, "(When second player connects, sh"... , 57) = 57
98 1674 17:21:29.245312 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
99 1674 17:21:29.770176 write(1, ".", 1) = 1
100 1674 17:21:29.770843 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
101 1674 17:21:30.271833 write(1, ".", 1) = 1
102 1674 17:21:30.272341 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
103 1674 17:21:30.784226 write(1, ".", 1) = 1
104 1674 17:21:30.784667 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
105 1674 17:21:31.286341 write(1, ".", 1) = 1
106 1674 17:21:31.287052 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
107 1674 17:21:31.797359 write(1, ".", 1) = 1
108 1674 17:21:31.797861 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
109 1674 17:21:32.305280 write(1, ".", 1) = 1
110 1674 17:21:32.305783 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
111 1674 17:21:32.814120 write(1, ".", 1) = 1
112 1674 17:21:32.814730 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
113 1674 17:21:33.320322 write(1, ".", 1) = 1
114 1674 17:21:33.321190 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
115 1674 17:21:33.825454 write(1, ".", 1) = 1
116 1674 17:21:33.826403 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,

```

```

    tv_nsec=500000000}, NULL) = 0
117 1674 17:21:34.330201 write(1, ".", 1) = 1
118 1674 17:21:34.331065 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
119 1674 17:21:34.835592 write(1, ".", 1) = 1
120 1674 17:21:34.836102 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
121 1674 17:21:35.340098 write(1, ".", 1) = 1
122 1674 17:21:35.340748 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
123 1674 17:21:35.844981 write(1, ".", 1) = 1
124 1674 17:21:35.845484 write(1, "\n", 1) = 1
125 1674 17:21:35.846172 write(1, "Second player connected!\n", 25) = 25
126 1674 17:21:35.846594 mmap(0x7f7fa8a9c000, 32864) = 0
127 1674 17:21:35.847152 close(3) = 0
128 1674 17:21:35.847633 write(1, "\n", 1) = 1
129 1674 17:21:35.848141 write(1, "===== GAME START ====="..., 33) = 33
130 1674 17:21:35.848598 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 3
131 1674 17:21:35.849120 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0)
    = 0x7f7fa8a9c000
132 1674 17:21:35.849511 write(1, "\n", 1) = 1
133 1674 17:21:35.849968 write(1, "===== CURRENT STATUS ====="..., 37) = 37
134 1674 17:21:35.850318 write(1, "\n", 1) = 1
135 1674 17:21:35.850582 write(1, "Your board:                Your sh"..., 37) = 37
136 1674 17:21:35.851049 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
137 1674 17:21:35.851383 write(1, " 0 . S . . . . . S S .  0 . . ."..., 48) = 48
138 1674 17:21:35.851706 write(1, " 1 . . . . . . . . . .  1 . . ."..., 48) = 48
139 1674 17:21:35.852167 write(1, " 2 S . . . . . S S . .  2 . . ."..., 48) = 48
140 1674 17:21:35.852720 write(1, " 3 . . . . . . . . . .  3 . . ."..., 48) = 48
141 1674 17:21:35.853205 write(1, " 4 S S S S . . . . . .  4 . . ."..., 48) = 48
142 1674 17:21:35.853659 write(1, " 5 . . . . . . S S . .  5 . . ."..., 48) = 48
143 1674 17:21:35.854122 write(1, " 6 . S . S . . . . . .  6 . . ."..., 48) = 48
144 1674 17:21:35.854502 write(1, " 7 . . . . . . . . . .  7 . . ."..., 48) = 48
145 1674 17:21:35.854900 write(1, " 8 S S S . . . S S S .  8 . . ."..., 48) = 48
146 1674 17:21:35.855300 write(1, " 9 . . . . . . . . . .  9 . . ."..., 48) = 48
147 1674 17:21:35.855616 write(1, "\n", 1) = 1
148 1674 17:21:35.855879 write(1, "=== YOUR TURN ===\n", 18) = 18
149 1674 17:21:35.856215 write(1, "Enter shot coordinates (row colu"..., 37) = 37
150 1674 17:21:35.856456 read(0, 0x558c9ac976b0, 1024) = ? ERESTARTSYS (To be
    restarted if SA_RESTART is set)
151 1674 17:21:36.930541 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
152 1674 17:21:36.930699 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
153 1674 17:21:36.930933 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
154 1674 17:21:36.931157 read(0, 0x558c9ac976b0, 1024) = ? ERESTARTSYS (To be
    restarted if SA_RESTART is set)
155 1674 17:21:37.047504 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
156 1674 17:21:37.047681 read(0, 0x558c9ac976b0, 1024) = ? ERESTARTSYS (To be
    restarted if SA_RESTART is set)
157 1674 17:21:37.111059 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
158 1674 17:21:37.111248 read(0, "0 22\n", 1024) = 5
159 1674 17:21:47.467630 write(1, "Error: coordinates out of bounds"..., 39) = 39

```

```

160 1674 17:21:47.468126 write(1, "\n", 1) = 1
161 1674 17:21:47.468590 write(1, "=====  
162 1674 17:21:47.468918 write(1, "\n", 1) = 1
163 1674 17:21:47.469208 write(1, "Your board:          Your sh"...  
164 1674 17:21:47.469533 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "...  
165 1674 17:21:47.469856 write(1, "  0 . S . . . . . S S .    0 . . . "...  
166 1674 17:21:47.470141 write(1, "  1 . . . . . . . . . .    1 . . . "...  
167 1674 17:21:47.470413 write(1, "  2 S . . . . . S S . .    2 . . . "...  
168 1674 17:21:47.470688 write(1, "  3 . . . . . . . . . .    3 . . . "...  
169 1674 17:21:47.470929 write(1, "  4 S S S S . . . . . .    4 . . . "...  
170 1674 17:21:47.471189 write(1, "  5 . . . . . . S S . .    5 . . . "...  
171 1674 17:21:47.471419 write(1, "  6 . S . S . . . . . .    6 . . . "...  
172 1674 17:21:47.471642 write(1, "  7 . . . . . . . . . .    7 . . . "...  
173 1674 17:21:47.471864 write(1, "  8 S S S . . . S S S .    8 . . . "...  
174 1674 17:21:47.472086 write(1, "  9 . . . . . . . . . .    9 . . . "...  
175 1674 17:21:47.472367 write(1, "\n", 1) = 1
176 1674 17:21:47.472607 write(1, "=== YOUR TURN ===\n", 18) = 18
177 1674 17:21:47.472822 write(1, "Enter shot coordinates (row colu"...  
178 1674 17:21:47.473008 read(0, "0 2\n", 1024) = 4
179 1674 17:21:48.647430 write(1, "...Sending shot to server...\n", 29) = 29
180 1674 17:21:48.647839 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
181 1674 17:21:48.649909 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
182 1674 17:21:48.651837 write(1, "...Waiting for server to process"...  
183 1674 17:21:48.652331 futex(0x7f7fa8a9c7f8,  
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808518,  
    tv_nsec=652259181}, FUTEX_BITSET_MATCH_ANY) = 0
184 1674 17:21:49.101021 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
185 1674 17:21:49.101955 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
186 1674 17:21:49.102442 write(1, "SUNK! You shoot again!\n", 23) = 23
187 1674 17:21:49.102752 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
188 1674 17:21:49.105133 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
189 1674 17:21:49.107340 write(1, "\n", 1) = 1
190 1674 17:21:49.107669 write(1, "=====  
191 1674 17:21:49.107994 write(1, "\n", 1) = 1
192 1674 17:21:49.108338 write(1, "Your board:          Your sh"...  
193 1674 17:21:49.108662 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "...  
194 1674 17:21:49.109117 write(1, "  0 . S . . . . . S S .    0 . . X"...  
195 1674 17:21:49.109529 write(1, "  1 . . . . . . . . . .    1 . . . "...  
196 1674 17:21:49.109926 write(1, "  2 S . . . . . S S . .    2 . . . "...  
197 1674 17:21:49.110362 write(1, "  3 . . . . . . . . . .    3 . . . "...  
198 1674 17:21:49.110762 write(1, "  4 S S S S . . . . . .    4 . . . "...  
199 1674 17:21:49.111298 write(1, "  5 . . . . . . S S . .    5 . . . "...  
200 1674 17:21:49.111774 write(1, "  6 . S . S . . . . . .    6 . . . "...  
201 1674 17:21:49.112059 write(1, "  7 . . . . . . . . . .    7 . . . "...  
202 1674 17:21:49.112400 write(1, "  8 S S S . . . S S S .    8 . . . "...  
203 1674 17:21:49.112767 write(1, "  9 . . . . . . . . . .    9 . . . "...  
204 1674 17:21:49.113101 write(1, "\n", 1) = 1
205 1674 17:21:49.113392 write(1, "=== YOUR TURN ===\n", 18) = 18
206 1674 17:21:49.113774 write(1, "Enter shot coordinates (row colu"...  
207 1674 17:21:49.114130 read(0, "1 5\n", 1024) = 4
208 1674 17:21:51.490802 write(1, "...Sending shot to server...\n", 29) = 29

```

```

209 1674 17:21:51.491262 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
210 1674 17:21:51.494255 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
211 1674 17:21:51.497407 write(1, "...Waiting for server to process"..., 41) = 41
212 1674 17:21:51.498022 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808521,
    tv_nsec=497938000}, FUTEX_BITSET_MATCH_ANY) = 0
213 1674 17:21:51.626626 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
214 1674 17:21:51.627706 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
215 1674 17:21:51.628047 write(1, "HIT! You shoot again!\n", 22) = 22
216 1674 17:21:51.628395 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
217 1674 17:21:51.630681 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
218 1674 17:21:51.631743 write(1, "\n", 1) = 1
219 1674 17:21:51.632163 write(1, "===== CURRENT STATUS ====="..., 37) = 37
220 1674 17:21:51.632578 write(1, "\n", 1) = 1
221 1674 17:21:51.633001 write(1, "Your board:                Your sh"..., 37) = 37
222 1674 17:21:51.633395 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
223 1674 17:21:51.633718 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
224 1674 17:21:51.634102 write(1, " 1 . . . . . . . . .   1 . . ."..., 48) = 48
225 1674 17:21:51.634528 write(1, " 2 S . . . . . S S .   2 . . ."..., 48) = 48
226 1674 17:21:51.634867 write(1, " 3 . . . . . . . . .   3 . . ."..., 48) = 48
227 1674 17:21:51.635178 write(1, " 4 S S S S . . . . .   4 . . ."..., 48) = 48
228 1674 17:21:51.635494 write(1, " 5 . . . . . . S S .   5 . . ."..., 48) = 48
229 1674 17:21:51.635863 write(1, " 6 . S . S . . . . .   6 . . ."..., 48) = 48
230 1674 17:21:51.636215 write(1, " 7 . . . . . . . . .   7 . . ."..., 48) = 48
231 1674 17:21:51.636550 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
232 1674 17:21:51.636900 write(1, " 9 . . . . . . . . .   9 . . ."..., 48) = 48
233 1674 17:21:51.637296 write(1, "\n", 1) = 1
234 1674 17:21:51.637797 write(1, "=== YOUR TURN ===\n", 18) = 18
235 1674 17:21:51.638204 write(1, "Enter shot coordinates (row colu"..., 37) = 37
236 1674 17:21:51.638515 read(0, "1 6\n", 1024) = 4
237 1674 17:21:52.640637 write(1, "...Sending shot to server...\n", 29) = 29
238 1674 17:21:52.640962 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
239 1674 17:21:52.643548 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
240 1674 17:21:52.645547 write(1, "...Waiting for server to process"..., 41) = 41
241 1674 17:21:52.646179 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808522,
    tv_nsec=646062199}, FUTEX_BITSET_MATCH_ANY) = 0
242 1674 17:21:53.142269 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
243 1674 17:21:53.143130 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
244 1674 17:21:53.143465 write(1, "SUNK! You shoot again!\n", 23) = 23
245 1674 17:21:53.143890 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
246 1674 17:21:53.146923 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
247 1674 17:21:53.148219 write(1, "\n", 1) = 1
248 1674 17:21:53.148686 write(1, "===== CURRENT STATUS ====="..., 37) = 37
249 1674 17:21:53.149070 write(1, "\n", 1) = 1
250 1674 17:21:53.149450 write(1, "Your board:                Your sh"..., 37) = 37
251 1674 17:21:53.149885 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
252 1674 17:21:53.150372 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
253 1674 17:21:53.150770 write(1, " 1 . . . . . . . . .   1 . . ."..., 48) = 48
254 1674 17:21:53.151129 write(1, " 2 S . . . . . S S .   2 . . ."..., 48) = 48
255 1674 17:21:53.151528 write(1, " 3 . . . . . . . . .   3 . . ."..., 48) = 48

```

```

256 1674 17:21:53.151901 write(1, " 4 S S S S . . . . . 4 . . ."..., 48) = 48
257 1674 17:21:53.152359 write(1, " 5 . . . . . S S . . 5 . . ."..., 48) = 48
258 1674 17:21:53.152915 write(1, " 6 . S . S . . . . . 6 . . ."..., 48) = 48
259 1674 17:21:53.153462 write(1, " 7 . . . . . . . . . 7 . . ."..., 48) = 48
260 1674 17:21:53.153808 write(1, " 8 S S S . . . S S S . 8 . . ."..., 48) = 48
261 1674 17:21:53.154200 write(1, " 9 . . . . . . . . . 9 . . ."..., 48) = 48
262 1674 17:21:53.154621 write(1, "\n", 1) = 1
263 1674 17:21:53.155082 write(1, "=== YOUR TURN ===\n", 18) = 18
264 1674 17:21:53.155516 write(1, "Enter shot coordinates (row colu"..., 37) = 37
265 1674 17:21:53.155959 read(0, "1 9\n", 1024) = 4
266 1674 17:21:54.684485 write(1, "...Sending shot to server...\n", 29) = 29
267 1674 17:21:54.684969 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
268 1674 17:21:54.687239 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
269 1674 17:21:54.689502 write(1, "...Waiting for server to process"..., 41) = 41
270 1674 17:21:54.690083 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808524,
    tv_nsec=689985316}, FUTEX_BITSET_MATCH_ANY) = 0
271 1674 17:21:55.177210 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
272 1674 17:21:55.177988 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
273 1674 17:21:55.178631 write(1, "HIT! You shoot again!\n", 22) = 22
274 1674 17:21:55.179097 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
275 1674 17:21:55.182045 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
276 1674 17:21:55.184015 write(1, "\n", 1) = 1
277 1674 17:21:55.184611 write(1, "===== CURRENT STATUS ====="..., 37) = 37
278 1674 17:21:55.185033 write(1, "\n", 1) = 1
279 1674 17:21:55.185390 write(1, "Your board:                Your sh"..., 37) = 37
280 1674 17:21:55.185746 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
281 1674 17:21:55.186062 write(1, " 0 . S . . . . . S S .  0 . . X"..., 48) = 48
282 1674 17:21:55.186435 write(1, " 1 . . . . . . . . .  1 . . ."..., 48) = 48
283 1674 17:21:55.186885 write(1, " 2 S . . . . . S S . .  2 . . ."..., 48) = 48
284 1674 17:21:55.187328 write(1, " 3 . . . . . . . . .  3 . . ."..., 48) = 48
285 1674 17:21:55.187823 write(1, " 4 S S S S . . . . .  4 . . ."..., 48) = 48
286 1674 17:21:55.188300 write(1, " 5 . . . . . . S S . .  5 . . ."..., 48) = 48
287 1674 17:21:55.188714 write(1, " 6 . S . S . . . . .  6 . . ."..., 48) = 48
288 1674 17:21:55.189090 write(1, " 7 . . . . . . . . .  7 . . ."..., 48) = 48
289 1674 17:21:55.189437 write(1, " 8 S S S . . . S S S .  8 . . ."..., 48) = 48
290 1674 17:21:55.189741 write(1, " 9 . . . . . . . . .  9 . . ."..., 48) = 48
291 1674 17:21:55.190060 write(1, "\n", 1) = 1
292 1674 17:21:55.190388 write(1, "=== YOUR TURN ===\n", 18) = 18
293 1674 17:21:55.190730 write(1, "Enter shot coordinates (row colu"..., 37) = 37
294 1674 17:21:55.191085 read(0, "2 9\n", 1024) = 4
295 1674 17:21:57.104380 write(1, "...Sending shot to server...\n", 29) = 29
296 1674 17:21:57.104785 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
297 1674 17:21:57.108098 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
298 1674 17:21:57.111502 write(1, "...Waiting for server to process"..., 41) = 41
299 1674 17:21:57.112153 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808527,
    tv_nsec=112015797}, FUTEX_BITSET_MATCH_ANY) = 0
300 1674 17:21:57.222503 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = -1 EAGAIN
    (Resource temporarily unavailable)
301 1674 17:21:57.222903 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0

```

```

302 1674 17:21:57.223250 write(1, "HIT! You shoot again!\n", 22) = 22
303 1674 17:21:57.223567 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
304 1674 17:21:57.226861 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
305 1674 17:21:57.228643 write(1, "\n", 1) = 1
306 1674 17:21:57.229254 write(1, "===== CURRENT STATUS =====", 37) = 37
307 1674 17:21:57.229689 write(1, "\n", 1) = 1
308 1674 17:21:57.230032 write(1, "Your board:          Your sh", 37) = 37
309 1674 17:21:57.230485 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 ", 46) = 46
310 1674 17:21:57.230893 write(1, "  0 . S . . . . . S S .    0 . . X", 48) = 48
311 1674 17:21:57.231345 write(1, "  1 . . . . . . . . . .    1 . . .", 48) = 48
312 1674 17:21:57.231715 write(1, "  2 S . . . . . S S . .    2 . . .", 48) = 48
313 1674 17:21:57.232071 write(1, "  3 . . . . . . . . . .    3 . . .", 48) = 48
314 1674 17:21:57.232472 write(1, "  4 S S S S . . . . . .    4 . . .", 48) = 48
315 1674 17:21:57.232873 write(1, "  5 . . . . . . S S . .    5 . . .", 48) = 48
316 1674 17:21:57.233226 write(1, "  6 . S . S . . . . . .    6 . . .", 48) = 48
317 1674 17:21:57.233515 write(1, "  7 . . . . . . . . . .    7 . . .", 48) = 48
318 1674 17:21:57.233796 write(1, "  8 S S S . . . S S S .    8 . . .", 48) = 48
319 1674 17:21:57.234066 write(1, "  9 . . . . . . . . . .    9 . . .", 48) = 48
320 1674 17:21:57.234375 write(1, "\n", 1) = 1
321 1674 17:21:57.234783 write(1, "=== YOUR TURN ===\n", 18) = 18
322 1674 17:21:57.235213 write(1, "Enter shot coordinates (row colu", 37) = 37
323 1674 17:21:57.235514 read(0, "3 1\n", 1024) = 4
324 1674 17:21:59.676857 write(1, "...Sending shot to server...\n", 29) = 29
325 1674 17:21:59.677221 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
326 1674 17:21:59.680248 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
327 1674 17:21:59.682245 write(1, "...Waiting for server to process", 41) = 41
328 1674 17:21:59.682779 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808529,
    tv_nsec=682667543}, FUTEX_BITSET_MATCH_ANY) = 0
329 1674 17:21:59.766936 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
330 1674 17:21:59.767895 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
331 1674 17:21:59.768229 write(1, "HIT! You shoot again!\n", 22) = 22
332 1674 17:21:59.768632 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
333 1674 17:21:59.772418 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
334 1674 17:21:59.773397 write(1, "\n", 1) = 1
335 1674 17:21:59.773812 write(1, "===== CURRENT STATUS =====", 37) = 37
336 1674 17:21:59.774270 write(1, "\n", 1) = 1
337 1674 17:21:59.774779 write(1, "Your board:          Your sh", 37) = 37
338 1674 17:21:59.775387 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 ", 46) = 46
339 1674 17:21:59.775847 write(1, "  0 . S . . . . . S S .    0 . . X", 48) = 48
340 1674 17:21:59.776232 write(1, "  1 . . . . . . . . . .    1 . . .", 48) = 48
341 1674 17:21:59.776557 write(1, "  2 S . . . . . S S . .    2 . . .", 48) = 48
342 1674 17:21:59.776940 write(1, "  3 . . . . . . . . . .    3 . X .", 48) = 48
343 1674 17:21:59.777250 write(1, "  4 S S S S . . . . . .    4 . . .", 48) = 48
344 1674 17:21:59.777574 write(1, "  5 . . . . . . S S . .    5 . . .", 48) = 48
345 1674 17:21:59.777931 write(1, "  6 . S . S . . . . . .    6 . . .", 48) = 48
346 1674 17:21:59.778224 write(1, "  7 . . . . . . . . . .    7 . . .", 48) = 48
347 1674 17:21:59.778556 write(1, "  8 S S S . . . S S S .    8 . . .", 48) = 48
348 1674 17:21:59.778961 write(1, "  9 . . . . . . . . . .    9 . . .", 48) = 48
349 1674 17:21:59.779250 write(1, "\n", 1) = 1
350 1674 17:21:59.779634 write(1, "=== YOUR TURN ===\n", 18) = 18

```

```

351 1674 17:21:59.780095 write(1, "Enter shot coordinates (row colu"..., 37) = 37
352 1674 17:21:59.780516 read(0, "3 3\n", 1024) = 4
353 1674 17:22:01.969502 write(1, "...Sending shot to server...\n", 29) = 29
354 1674 17:22:01.970007 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
355 1674 17:22:01.973048 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
356 1674 17:22:01.974957 write(1, "...Waiting for server to process"..., 41) = 41
357 1674 17:22:01.975442 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808531,
    tv_nsec=975366261}, FUTEX_BITSET_MATCH_ANY) = 0
358 1674 17:22:02.317633 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
359 1674 17:22:02.318393 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
360 1674 17:22:02.318814 write(1, "SUNK! You shoot again!\n", 23) = 23
361 1674 17:22:02.319213 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
362 1674 17:22:02.323251 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
363 1674 17:22:02.324406 write(1, "\n", 1) = 1
364 1674 17:22:02.324971 write(1, "===== CURRENT STATUS ====="..., 37) = 37
365 1674 17:22:02.325317 write(1, "\n", 1) = 1
366 1674 17:22:02.325684 write(1, "Your board:                Your sh"..., 37) = 37
367 1674 17:22:02.326067 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
368 1674 17:22:02.326565 write(1, "  0 . S . . . . . S S .  0 . . X"..., 48) = 48
369 1674 17:22:02.327004 write(1, "  1 . . . . . . . . . .  1 . . ."..., 48) = 48
370 1674 17:22:02.327382 write(1, "  2 S . . . . . S S . .  2 . . ."..., 48) = 48
371 1674 17:22:02.327789 write(1, "  3 . . . . . . . . . .  3 . X ."..., 48) = 48
372 1674 17:22:02.328211 write(1, "  4 S S S S . . . . . .  4 . . ."..., 48) = 48
373 1674 17:22:02.328612 write(1, "  5 . . . . . . S S . .  5 . . ."..., 48) = 48
374 1674 17:22:02.328937 write(1, "  6 . S . S . . . . . .  6 . . ."..., 48) = 48
375 1674 17:22:02.329259 write(1, "  7 . . . . . . . . . .  7 . . ."..., 48) = 48
376 1674 17:22:02.329718 write(1, "  8 S S S . . . S S S .  8 . . ."..., 48) = 48
377 1674 17:22:02.330117 write(1, "  9 . . . . . . . . . .  9 . . ."..., 48) = 48
378 1674 17:22:02.330491 write(1, "\n", 1) = 1
379 1674 17:22:02.330928 write(1, "=== YOUR TURN ===\n", 18) = 18
380 1674 17:22:02.331404 write(1, "Enter shot coordinates (row colu"..., 37) = 37
381 1674 17:22:02.331852 read(0, "3 9\n", 1024) = 4
382 1674 17:22:05.000937 write(1, "...Sending shot to server...\n", 29) = 29
383 1674 17:22:05.001299 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
384 1674 17:22:05.003325 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
385 1674 17:22:05.005198 write(1, "...Waiting for server to process"..., 41) = 41
386 1674 17:22:05.005678 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808535,
    tv_nsec=5603421}, FUTEX_BITSET_MATCH_ANY) = 0
387 1674 17:22:05.399313 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
388 1674 17:22:05.400139 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
389 1674 17:22:05.400451 write(1, "SUNK! You shoot again!\n", 23) = 23
390 1674 17:22:05.400761 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
391 1674 17:22:05.403377 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
392 1674 17:22:05.406068 write(1, "\n", 1) = 1
393 1674 17:22:05.406644 write(1, "===== CURRENT STATUS ====="..., 37) = 37
394 1674 17:22:05.407191 write(1, "\n", 1) = 1
395 1674 17:22:05.407552 write(1, "Your board:                Your sh"..., 37) = 37
396 1674 17:22:05.407935 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
397 1674 17:22:05.408503 write(1, "  0 . S . . . . . S S .  0 . . X"..., 48) = 48

```

```

398 1674 17:22:05.409005 write(1, " 1 . . . . . 1 . . ."..., 48) = 48
399 1674 17:22:05.409380 write(1, " 2 S . . . . . S S . . 2 . . ."..., 48) = 48
400 1674 17:22:05.409687 write(1, " 3 . . . . . 3 . X ."..., 48) = 48
401 1674 17:22:05.410012 write(1, " 4 S S S S . . . . . 4 . . ."..., 48) = 48
402 1674 17:22:05.410410 write(1, " 5 . . . . . S S . . 5 . . ."..., 48) = 48
403 1674 17:22:05.410729 write(1, " 6 . S . S . . . . . 6 . . ."..., 48) = 48
404 1674 17:22:05.411047 write(1, " 7 . . . . . 7 . . ."..., 48) = 48
405 1674 17:22:05.411331 write(1, " 8 S S S . . . S S S . 8 . . ."..., 48) = 48
406 1674 17:22:05.411730 write(1, " 9 . . . . . 9 . . ."..., 48) = 48
407 1674 17:22:05.412163 write(1, "\n", 1) = 1
408 1674 17:22:05.412527 write(1, "=== YOUR TURN ===\n", 18) = 18
409 1674 17:22:05.412950 write(1, "Enter shot coordinates (row colu"..., 37) = 37
410 1674 17:22:05.413386 read(0, "4 1\n", 1024) = 4
411 1674 17:22:08.080865 write(1, "...Sending shot to server...\n", 29) = 29
412 1674 17:22:08.081239 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
413 1674 17:22:08.083389 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
414 1674 17:22:08.087051 write(1, "...Waiting for server to process"..., 41) = 41
415 1674 17:22:08.087611 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808538,
    tv_nsec=87522843}, FUTEX_BITSET_MATCH_ANY) = 0
416 1674 17:22:08.509651 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
417 1674 17:22:08.510595 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
418 1674 17:22:08.511096 write(1, "SUNK! You shoot again!\n", 23) = 23
419 1674 17:22:08.511558 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
420 1674 17:22:08.513634 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
421 1674 17:22:08.514825 write(1, "\n", 1) = 1
422 1674 17:22:08.515344 write(1, "===== CURRENT STATUS ====="..., 37) = 37
423 1674 17:22:08.515849 write(1, "\n", 1) = 1
424 1674 17:22:08.516262 write(1, "Your board:                Your sh"..., 37) = 37
425 1674 17:22:08.516752 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
426 1674 17:22:08.517169 write(1, " 0 . S . . . . . S S .  0 . . X"..., 48) = 48
427 1674 17:22:08.517587 write(1, " 1 . . . . . 1 . . ."..., 48) = 48
428 1674 17:22:08.518032 write(1, " 2 S . . . . . S S . . 2 . . ."..., 48) = 48
429 1674 17:22:08.518347 write(1, " 3 . . . . . 3 . X ."..., 48) = 48
430 1674 17:22:08.518661 write(1, " 4 S S S S . . . . . 4 . X ."..., 48) = 48
431 1674 17:22:08.518954 write(1, " 5 . . . . . S S . . 5 . . ."..., 48) = 48
432 1674 17:22:08.519398 write(1, " 6 . S . S . . . . . 6 . . ."..., 48) = 48
433 1674 17:22:08.519757 write(1, " 7 . . . . . 7 . . ."..., 48) = 48
434 1674 17:22:08.520039 write(1, " 8 S S S . . . S S S .  8 . . ."..., 48) = 48
435 1674 17:22:08.520343 write(1, " 9 . . . . . 9 . . ."..., 48) = 48
436 1674 17:22:08.520623 write(1, "\n", 1) = 1
437 1674 17:22:08.520850 write(1, "=== YOUR TURN ===\n", 18) = 18
438 1674 17:22:08.521248 write(1, "Enter shot coordinates (row colu"..., 37) = 37
439 1674 17:22:08.521580 read(0, "5 6\n", 1024) = 4
440 1674 17:22:11.232475 write(1, "...Sending shot to server...\n", 29) = 29
441 1674 17:22:11.232935 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
442 1674 17:22:11.235632 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
443 1674 17:22:11.239726 write(1, "...Waiting for server to process"..., 41) = 41
444 1674 17:22:11.240276 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808541,
    tv_nsec=240205477}, FUTEX_BITSET_MATCH_ANY) = 0

```

```

445 1674 17:22:11.538062 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
446 1674 17:22:11.538732 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
447 1674 17:22:11.538985 write(1, "HIT! You shoot again!\n", 22) = 22
448 1674 17:22:11.539264 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
449 1674 17:22:11.541038 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
450 1674 17:22:11.542097 write(1, "\n", 1) = 1
451 1674 17:22:11.542503 write(1, "=====CURRENT STATUS====="..., 37) = 37
452 1674 17:22:11.542833 write(1, "\n", 1) = 1
453 1674 17:22:11.543214 write(1, "Your board:                Your sh"..., 37) = 37
454 1674 17:22:11.543532 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
455 1674 17:22:11.543851 write(1, " 0 . S . . . . . S S .  0 . . X"..., 48) = 48
456 1674 17:22:11.544154 write(1, " 1 . . . . . . . . . .  1 . . ."..., 48) = 48
457 1674 17:22:11.544452 write(1, " 2 S . . . . . S S . .  2 . . ."..., 48) = 48
458 1674 17:22:11.544726 write(1, " 3 . . . . . . . . . .  3 . X ."..., 48) = 48
459 1674 17:22:11.544973 write(1, " 4 S S S S . . . . . .  4 . X ."..., 48) = 48
460 1674 17:22:11.545222 write(1, " 5 . . . . . . S S . .  5 . . ."..., 48) = 48
461 1674 17:22:11.545458 write(1, " 6 . S . S . . . . . .  6 . . ."..., 48) = 48
462 1674 17:22:11.545748 write(1, " 7 . . . . . . . . . .  7 . . ."..., 48) = 48
463 1674 17:22:11.546066 write(1, " 8 S S S . . . S S S .  8 . . ."..., 48) = 48
464 1674 17:22:11.546354 write(1, " 9 . . . . . . . . . .  9 . . ."..., 48) = 48
465 1674 17:22:11.546653 write(1, "\n", 1) = 1
466 1674 17:22:11.546979 write(1, "=== YOUR TURN ===\n", 18) = 18
467 1674 17:22:11.547320 write(1, "Enter shot coordinates (row colu"..., 37) = 37
468 1674 17:22:11.547572 read(0, "5 7\n", 1024) = 4
469 1674 17:22:13.458658 write(1, "...Sending shot to server...\n", 29) = 29
470 1674 17:22:13.459077 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
471 1674 17:22:13.461724 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
472 1674 17:22:13.463801 write(1, "...Waiting for server to process"..., 41) = 41
473 1674 17:22:13.464231 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808543,
    tv_nsec=464125400}, FUTEX_BITSET_MATCH_ANY) = 0
474 1674 17:22:13.566178 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
475 1674 17:22:13.566740 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
476 1674 17:22:13.567046 write(1, "SUNK! You shoot again!\n", 23) = 23
477 1674 17:22:13.567415 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
478 1674 17:22:13.569231 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
479 1674 17:22:13.570403 write(1, "\n", 1) = 1
480 1674 17:22:13.570739 write(1, "=====CURRENT STATUS====="..., 37) = 37
481 1674 17:22:13.571051 write(1, "\n", 1) = 1
482 1674 17:22:13.571358 write(1, "Your board:                Your sh"..., 37) = 37
483 1674 17:22:13.571627 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
484 1674 17:22:13.572044 write(1, " 0 . S . . . . . S S .  0 . . X"..., 48) = 48
485 1674 17:22:13.572437 write(1, " 1 . . . . . . . . . .  1 . . ."..., 48) = 48
486 1674 17:22:13.572781 write(1, " 2 S . . . . . S S . .  2 . . ."..., 48) = 48
487 1674 17:22:13.573152 write(1, " 3 . . . . . . . . . .  3 . X ."..., 48) = 48
488 1674 17:22:13.573435 write(1, " 4 S S S S . . . . . .  4 . X ."..., 48) = 48
489 1674 17:22:13.573709 write(1, " 5 . . . . . . S S . .  5 . . ."..., 48) = 48
490 1674 17:22:13.573955 write(1, " 6 . S . S . . . . . .  6 . . ."..., 48) = 48
491 1674 17:22:13.574182 write(1, " 7 . . . . . . . . . .  7 . . ."..., 48) = 48
492 1674 17:22:13.574451 write(1, " 8 S S S . . . S S S .  8 . . ."..., 48) = 48
493 1674 17:22:13.574716 write(1, " 9 . . . . . . . . . .  9 . . ."..., 48) = 48

```

```

494 1674 17:22:13.575066 write(1, "\n", 1) = 1
495 1674 17:22:13.575336 write(1, "=== YOUR TURN ===\n", 18) = 18
496 1674 17:22:13.575614 write(1, "Enter shot coordinates (row colu"... , 37) = 37
497 1674 17:22:13.575879 read(0, "6 \n", 1024) = 3
498 1674 17:22:16.496183 read(0, "1\n", 1024) = 2
499 1674 17:22:18.098781 write(1, "...Sending shot to server...\n", 29) = 29
500 1674 17:22:18.099488 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
501 1674 17:22:18.102169 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
502 1674 17:22:18.104279 write(1, "...Waiting for server to process"... , 41) = 41
503 1674 17:22:18.104861 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808548,
    tv_nsec=104821034}, FUTEX_BITSET_MATCH_ANY) = 0
504 1674 17:22:18.152230 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
505 1674 17:22:18.153180 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
506 1674 17:22:18.153385 write(1, "HIT! You shoot again!\n", 22) = 22
507 1674 17:22:18.153638 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
508 1674 17:22:18.155348 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
509 1674 17:22:18.158542 write(1, "\n", 1) = 1
510 1674 17:22:18.159245 write(1, "===== CURRENT STATUS ====="... , 37) = 37
511 1674 17:22:18.159920 write(1, "\n", 1) = 1
512 1674 17:22:18.160295 write(1, "Your board:                Your sh"... , 37) = 37
513 1674 17:22:18.160632 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "... , 46) = 46
514 1674 17:22:18.161012 write(1, " 0 . S . . . . . S S .   0 . . X"... , 48) = 48
515 1674 17:22:18.161327 write(1, " 1 . . . . . . . . . .   1 . . ."... , 48) = 48
516 1674 17:22:18.161614 write(1, " 2 S . . . . . S S . .   2 . . ."... , 48) = 48
517 1674 17:22:18.161930 write(1, " 3 . . . . . . . . . .   3 . X ."... , 48) = 48
518 1674 17:22:18.162248 write(1, " 4 S S S S . . . . . .   4 . X ."... , 48) = 48
519 1674 17:22:18.162627 write(1, " 5 . . . . . . S S . .   5 . . ."... , 48) = 48
520 1674 17:22:18.162868 write(1, " 6 . S . S . . . . . .   6 . X ."... , 48) = 48
521 1674 17:22:18.163118 write(1, " 7 . . . . . . . . . .   7 . . ."... , 48) = 48
522 1674 17:22:18.163330 write(1, " 8 S S S . . . S S S .   8 . . ."... , 48) = 48
523 1674 17:22:18.163529 write(1, " 9 . . . . . . . . . .   9 . . ."... , 48) = 48
524 1674 17:22:18.163862 write(1, "\n", 1) = 1
525 1674 17:22:18.164152 write(1, "=== YOUR TURN ===\n", 18) = 18
526 1674 17:22:18.164444 write(1, "Enter shot coordinates (row colu"... , 37) = 37
527 1674 17:22:18.164718 read(0, "6 2\n", 1024) = 4
528 1674 17:22:21.036839 write(1, "...Sending shot to server...\n", 29) = 29
529 1674 17:22:21.037157 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
530 1674 17:22:21.039935 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
531 1674 17:22:21.043341 write(1, "...Waiting for server to process"... , 41) = 41
532 1674 17:22:21.043945 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808551,
    tv_nsec=43818691}, FUTEX_BITSET_MATCH_ANY) = 0
533 1674 17:22:21.398996 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
534 1674 17:22:21.400002 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
535 1674 17:22:21.400319 write(1, "HIT! You shoot again!\n", 22) = 22
536 1674 17:22:21.400669 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
537 1674 17:22:21.403403 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
538 1674 17:22:21.404452 write(1, "\n", 1) = 1
539 1674 17:22:21.404887 write(1, "===== CURRENT STATUS ====="... , 37) = 37
540 1674 17:22:21.405277 write(1, "\n", 1) = 1

```

```

541 1674 17:22:21.405594 write(1, "Your board:          Your sh"... , 37) = 37
542 1674 17:22:21.405917 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "... , 46) = 46
543 1674 17:22:21.406189 write(1, "  0 . S . . . . . S S .    0 . . X"... , 48) = 48
544 1674 17:22:21.406549 write(1, "  1 . . . . . . . . . .    1 . . ."... , 48) = 48
545 1674 17:22:21.406987 write(1, "  2 S . . . . . S S . .    2 . . ."... , 48) = 48
546 1674 17:22:21.407322 write(1, "  3 . . . . . . . . . .    3 . X ."... , 48) = 48
547 1674 17:22:21.407669 write(1, "  4 S S S S . . . . . .    4 . X ."... , 48) = 48
548 1674 17:22:21.408013 write(1, "  5 . . . . . . S S . .    5 . . ."... , 48) = 48
549 1674 17:22:21.408333 write(1, "  6 . S . S . . . . . .    6 . X X"... , 48) = 48
550 1674 17:22:21.408690 write(1, "  7 . . . . . . . . . .    7 . . ."... , 48) = 48
551 1674 17:22:21.409068 write(1, "  8 S S S . . . S S S .    8 . . ."... , 48) = 48
552 1674 17:22:21.409507 write(1, "  9 . . . . . . . . . .    9 . . ."... , 48) = 48
553 1674 17:22:21.409862 write(1, "\n", 1) = 1
554 1674 17:22:21.410228 write(1, "=== YOUR TURN ===\n", 18) = 18
555 1674 17:22:21.410627 write(1, "Enter shot coordinates (row colu"... , 37) = 37
556 1674 17:22:21.411014 read(0, "6 3\n", 1024) = 4
557 1674 17:22:23.285405 write(1, "...Sending shot to server...\n", 29) = 29
558 1674 17:22:23.285855 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
559 1674 17:22:23.287912 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
560 1674 17:22:23.289900 write(1, "...Waiting for server to process"... , 41) = 41
561 1674 17:22:23.290414 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808553,
    tv_nsec=290338363}, FUTEX_BITSET_MATCH_ANY) = 0
562 1674 17:22:23.424476 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
563 1674 17:22:23.425468 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
564 1674 17:22:23.425749 write(1, "SUNK! You shoot again!\n", 23) = 23
565 1674 17:22:23.426207 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
566 1674 17:22:23.428192 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
567 1674 17:22:23.430646 write(1, "\n", 1) = 1
568 1674 17:22:23.431211 write(1, "===== CURRENT STATUS ====="... , 37) = 37
569 1674 17:22:23.431818 write(1, "\n", 1) = 1
570 1674 17:22:23.432414 write(1, "Your board:          Your sh"... , 37) = 37
571 1674 17:22:23.432882 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "... , 46) = 46
572 1674 17:22:23.433225 write(1, "  0 . S . . . . . S S .    0 . . X"... , 48) = 48
573 1674 17:22:23.433612 write(1, "  1 . . . . . . . . . .    1 . . ."... , 48) = 48
574 1674 17:22:23.433896 write(1, "  2 S . . . . . S S . .    2 . . ."... , 48) = 48
575 1674 17:22:23.434233 write(1, "  3 . . . . . . . . . .    3 . X ."... , 48) = 48
576 1674 17:22:23.434579 write(1, "  4 S S S S . . . . . .    4 . X ."... , 48) = 48
577 1674 17:22:23.434999 write(1, "  5 . . . . . . S S . .    5 . . ."... , 48) = 48
578 1674 17:22:23.435410 write(1, "  6 . S . S . . . . . .    6 . X X"... , 48) = 48
579 1674 17:22:23.435738 write(1, "  7 . . . . . . . . . .    7 . . ."... , 48) = 48
580 1674 17:22:23.436026 write(1, "  8 S S S . . . S S S .    8 . . ."... , 48) = 48
581 1674 17:22:23.436364 write(1, "  9 . . . . . . . . . .    9 . . ."... , 48) = 48
582 1674 17:22:23.436642 write(1, "\n", 1) = 1
583 1674 17:22:23.436980 write(1, "=== YOUR TURN ===\n", 18) = 18
584 1674 17:22:23.437349 write(1, "Enter shot coordinates (row colu"... , 37) = 37
585 1674 17:22:23.437672 read(0, "7 8\n", 1024) = 4
586 1674 17:22:26.067841 write(1, "...Sending shot to server...\n", 29) = 29
587 1674 17:22:26.068343 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
588 1674 17:22:26.071208 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
589 1674 17:22:26.075055 write(1, "...Waiting for server to process"... , 41) = 41

```

```

590 1674 17:22:26.075617 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808556,
    tv_nsec=75534340}, FUTEX_BITSET_MATCH_ANY) = 0
591 1674 17:22:26.469695 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
592 1674 17:22:26.470510 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
593 1674 17:22:26.471013 write(1, "SUNK! You shoot again!\n", 23) = 23
594 1674 17:22:26.471402 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
595 1674 17:22:26.474819 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
596 1674 17:22:26.476623 write(1, "\n", 1) = 1
597 1674 17:22:26.477298 write(1, "===== CURRENT STATUS =====", 37) = 37
598 1674 17:22:26.477717 write(1, "\n", 1) = 1
599 1674 17:22:26.478074 write(1, "Your board:          Your sh", 37) = 37
600 1674 17:22:26.478481 write(1, "  0 1 2 3 4 5 6 7 8 9  0 1 2 ", 46) = 46
601 1674 17:22:26.478829 write(1, "  0 . S . . . . . S S .  0 . . X", 48) = 48
602 1674 17:22:26.479197 write(1, "  1 . . . . . . . . . .  1 . . .", 48) = 48
603 1674 17:22:26.479589 write(1, "  2 S . . . . . S S . .  2 . . .", 48) = 48
604 1674 17:22:26.480064 write(1, "  3 . . . . . . . . . .  3 . X .", 48) = 48
605 1674 17:22:26.480569 write(1, "  4 S S S S . . . . . .  4 . X .", 48) = 48
606 1674 17:22:26.480983 write(1, "  5 . . . . . . S S . .  5 . . .", 48) = 48
607 1674 17:22:26.481327 write(1, "  6 . S . S . . . . . .  6 . X X", 48) = 48
608 1674 17:22:26.481675 write(1, "  7 . . . . . . . . . .  7 . . .", 48) = 48
609 1674 17:22:26.482029 write(1, "  8 S S S . . . S S S .  8 . . .", 48) = 48
610 1674 17:22:26.482482 write(1, "  9 . . . . . . . . . .  9 . . .", 48) = 48
611 1674 17:22:26.482867 write(1, "\n", 1) = 1
612 1674 17:22:26.483231 write(1, "=== YOUR TURN ===\n", 18) = 18
613 1674 17:22:26.483627 write(1, "Enter shot coordinates (row colu", 37) = 37
614 1674 17:22:26.484008 read(0, "8 0\n", 1024) = 4
615 1674 17:22:29.083250 write(1, "...Sending shot to server...\n", 29) = 29
616 1674 17:22:29.083797 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
617 1674 17:22:29.086509 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
618 1674 17:22:29.088611 write(1, "...Waiting for server to process", 41) = 41
619 1674 17:22:29.089082 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808559,
    tv_nsec=89016831}, FUTEX_BITSET_MATCH_ANY) = 0
620 1674 17:22:29.526579 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
621 1674 17:22:29.527522 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
622 1674 17:22:29.527947 write(1, "SUNK! You shoot again!\n", 23) = 23
623 1674 17:22:29.528347 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
624 1674 17:22:29.530610 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
625 1674 17:22:29.531309 write(1, "\n", 1) = 1
626 1674 17:22:29.531784 write(1, "===== CURRENT STATUS =====", 37) = 37
627 1674 17:22:29.532236 write(1, "\n", 1) = 1
628 1674 17:22:29.532664 write(1, "Your board:          Your sh", 37) = 37
629 1674 17:22:29.532939 write(1, "  0 1 2 3 4 5 6 7 8 9  0 1 2 ", 46) = 46
630 1674 17:22:29.533315 write(1, "  0 . S . . . . . S S .  0 . . X", 48) = 48
631 1674 17:22:29.533636 write(1, "  1 . . . . . . . . . .  1 . . .", 48) = 48
632 1674 17:22:29.533905 write(1, "  2 S . . . . . S S . .  2 . . .", 48) = 48
633 1674 17:22:29.534204 write(1, "  3 . . . . . . . . . .  3 . X .", 48) = 48
634 1674 17:22:29.534621 write(1, "  4 S S S S . . . . . .  4 . X .", 48) = 48
635 1674 17:22:29.534994 write(1, "  5 . . . . . . S S . .  5 . . .", 48) = 48
636 1674 17:22:29.535462 write(1, "  6 . S . S . . . . . .  6 . X X", 48) = 48

```

```

637 1674 17:22:29.535806 write(1, " 7 . . . . . 7 . . ."..., 48) = 48
638 1674 17:22:29.536027 write(1, " 8 S S S . . . S S S . 8 X . ."..., 48) = 48
639 1674 17:22:29.536323 write(1, " 9 . . . . . 9 . . ."..., 48) = 48
640 1674 17:22:29.536696 write(1, "\n", 1) = 1
641 1674 17:22:29.537009 write(1, "=== YOUR TURN ===\n", 18) = 18
642 1674 17:22:29.537324 write(1, "Enter shot coordinates (row colu"..., 37) = 37
643 1674 17:22:29.537641 read(0, "9 5\n", 1024) = 4
644 1674 17:22:34.738858 write(1, "...Sending shot to server...\n", 29) = 29
645 1674 17:22:34.739472 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
646 1674 17:22:34.742386 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
647 1674 17:22:34.745890 write(1, "...Waiting for server to process"..., 41) = 41
648 1674 17:22:34.746461 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808564,
    tv_nsec=746341641}, FUTEX_BITSET_MATCH_ANY) = 0
649 1674 17:22:35.167875 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
650 1674 17:22:35.168648 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
651 1674 17:22:35.168931 write(1, "HIT! You shoot again!\n", 22) = 22
652 1674 17:22:35.169246 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
653 1674 17:22:35.171104 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
654 1674 17:22:35.172197 write(1, "\n", 1) = 1
655 1674 17:22:35.172541 write(1, "===== CURRENT STATUS ====="..., 37) = 37
656 1674 17:22:35.172847 write(1, "\n", 1) = 1
657 1674 17:22:35.173245 write(1, "Your board:                Your sh"..., 37) = 37
658 1674 17:22:35.173695 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
659 1674 17:22:35.174173 write(1, " 0 . S . . . . . S S .  0 . . X"..., 48) = 48
660 1674 17:22:35.174669 write(1, " 1 . . . . . 1 . . ."..., 48) = 48
661 1674 17:22:35.175196 write(1, " 2 S . . . . . S S .  2 . . ."..., 48) = 48
662 1674 17:22:35.175671 write(1, " 3 . . . . . 3 . X ."..., 48) = 48
663 1674 17:22:35.176151 write(1, " 4 S S S S . . . . . 4 . X ."..., 48) = 48
664 1674 17:22:35.176602 write(1, " 5 . . . . . S S .  5 . . ."..., 48) = 48
665 1674 17:22:35.177013 write(1, " 6 . S . S . . . . . 6 . X X"..., 48) = 48
666 1674 17:22:35.177355 write(1, " 7 . . . . . 7 . . ."..., 48) = 48
667 1674 17:22:35.177727 write(1, " 8 S S S . . . S S S . 8 X . ."..., 48) = 48
668 1674 17:22:35.178046 write(1, " 9 . . . . . 9 . . ."..., 48) = 48
669 1674 17:22:35.178324 write(1, "\n", 1) = 1
670 1674 17:22:35.178583 write(1, "=== YOUR TURN ===\n", 18) = 18
671 1674 17:22:35.178938 write(1, "Enter shot coordinates (row colu"..., 37) = 37
672 1674 17:22:35.179223 read(0, "9 5\n", 1024) = 4
673 1674 17:22:36.217772 write(1, "Error: you already shot here!\n", 30) = 30
674 1674 17:22:36.218194 write(1, "\n", 1) = 1
675 1674 17:22:36.218551 write(1, "===== CURRENT STATUS ====="..., 37) = 37
676 1674 17:22:36.218879 write(1, "\n", 1) = 1
677 1674 17:22:36.219282 write(1, "Your board:                Your sh"..., 37) = 37
678 1674 17:22:36.219751 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
679 1674 17:22:36.220092 write(1, " 0 . S . . . . . S S .  0 . . X"..., 48) = 48
680 1674 17:22:36.220538 write(1, " 1 . . . . . 1 . . ."..., 48) = 48
681 1674 17:22:36.221015 write(1, " 2 S . . . . . S S .  2 . . ."..., 48) = 48
682 1674 17:22:36.221402 write(1, " 3 . . . . . 3 . X ."..., 48) = 48
683 1674 17:22:36.221842 write(1, " 4 S S S S . . . . . 4 . X ."..., 48) = 48
684 1674 17:22:36.222229 write(1, " 5 . . . . . S S .  5 . . ."..., 48) = 48
685 1674 17:22:36.222600 write(1, " 6 . S . S . . . . . 6 . X X"..., 48) = 48

```

```

686 1674 17:22:36.222927 write(1, " 7 . . . . . 7 . . ."..., 48) = 48
687 1674 17:22:36.223283 write(1, " 8 S S S . . . S S S . 8 X . ."..., 48) = 48
688 1674 17:22:36.223646 write(1, " 9 . . . . . 9 . . ."..., 48) = 48
689 1674 17:22:36.223977 write(1, "\n", 1) = 1
690 1674 17:22:36.224332 write(1, "=== YOUR TURN ===\n", 18) = 18
691 1674 17:22:36.224583 write(1, "Enter shot coordinates (row colu"..., 37) = 37
692 1674 17:22:36.224943 read(0, "9 4\n", 1024) = 4
693 1674 17:22:38.519949 write(1, "...Sending shot to server...\n", 29) = 29
694 1674 17:22:38.520430 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
695 1674 17:22:38.523235 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
696 1674 17:22:38.525860 write(1, "...Waiting for server to process"..., 41) = 41
697 1674 17:22:38.526484 futex(0x7f7fa8a9c7fc,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808568,
    tv_nsec=526377717}, FUTEX_BITSET_MATCH_ANY) = 0
698 1674 17:22:38.754805 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = -1 EAGAIN
    (Resource temporarily unavailable)
699 1674 17:22:38.755253 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
700 1674 17:22:38.755656 write(1, "HIT! You shoot again!\n", 22) = 22
701 1674 17:22:38.756007 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
702 1674 17:22:38.757856 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
703 1674 17:22:38.758757 write(1, "\n", 1) = 1
704 1674 17:22:38.759087 write(1, "===== CURRENT STATUS ====="..., 37) = 37
705 1674 17:22:38.759388 write(1, "\n", 1) = 1
706 1674 17:22:38.759620 write(1, "Your board:                Your sh"..., 37) = 37
707 1674 17:22:38.759878 write(1, "  0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
708 1674 17:22:38.760074 write(1, " 0 . S . . . . . S S .  0 . . X"..., 48) = 48
709 1674 17:22:38.760525 write(1, " 1 . . . . . 1 . . ."..., 48) = 48
710 1674 17:22:38.761034 write(1, " 2 S . . . . . S S .  2 . . ."..., 48) = 48
711 1674 17:22:38.761440 write(1, " 3 . . . . . 3 . X ."..., 48) = 48
712 1674 17:22:38.761728 write(1, " 4 S S S S . . . . . 4 . X ."..., 48) = 48
713 1674 17:22:38.762012 write(1, " 5 . . . . . S S .  5 . . ."..., 48) = 48
714 1674 17:22:38.762335 write(1, " 6 . S . S . . . . . 6 . X X"..., 48) = 48
715 1674 17:22:38.762672 write(1, " 7 . . . . . 7 . . ."..., 48) = 48
716 1674 17:22:38.762998 write(1, " 8 S S S . . . S S S .  8 X . ."..., 48) = 48
717 1674 17:22:38.763290 write(1, " 9 . . . . . 9 . . ."..., 48) = 48
718 1674 17:22:38.763577 write(1, "\n", 1) = 1
719 1674 17:22:38.763957 write(1, "=== YOUR TURN ===\n", 18) = 18
720 1674 17:22:38.764346 write(1, "Enter shot coordinates (row colu"..., 37) = 37
721 1674 17:22:38.764692 read(0, "9 6\n", 1024) = 4
722 1674 17:22:40.385710 write(1, "...Sending shot to server...\n", 29) = 29
723 1674 17:22:40.386154 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
724 1674 17:22:40.388490 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
725 1674 17:22:40.390299 write(1, "...Waiting for server to process"..., 41) = 41
726 1674 17:22:40.390951 futex(0x7f7fa8a9c7f8,
    FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808570,
    tv_nsec=390864077}, FUTEX_BITSET_MATCH_ANY) = 0
727 1674 17:22:40.809380 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = -1 EAGAIN
    (Resource temporarily unavailable)
728 1674 17:22:40.809718 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
729 1674 17:22:40.809943 write(1, "HIT! You shoot again!\n", 22) = 22
730 1674 17:22:40.810180 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0

```

```

731 1674 17:22:40.813172 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
732 1674 17:22:40.814164 write(1, "\n", 1) = 1
733 1674 17:22:40.814456 write(1, "===== CURRENT STATUS ====="..., 37) = 37
734 1674 17:22:40.814814 write(1, "\n", 1) = 1
735 1674 17:22:40.815128 write(1, "Your board:          Your sh"..., 37) = 37
736 1674 17:22:40.815446 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
737 1674 17:22:40.815746 write(1, "  0 . S . . . . . S S .   0 . . X"..., 48) = 48
738 1674 17:22:40.816025 write(1, "  1 . . . . . . . . . .   1 . . ."..., 48) = 48
739 1674 17:22:40.816319 write(1, "  2 S . . . . . S S . .   2 . . ."..., 48) = 48
740 1674 17:22:40.816626 write(1, "  3 . . . . . . . . . .   3 . X ."..., 48) = 48
741 1674 17:22:40.816993 write(1, "  4 S S S S . . . . .     4 . X ."..., 48) = 48
742 1674 17:22:40.817296 write(1, "  5 . . . . . . S S . .   5 . . ."..., 48) = 48
743 1674 17:22:40.817538 write(1, "  6 . S . S . . . . .     6 . X X"..., 48) = 48
744 1674 17:22:40.817792 write(1, "  7 . . . . . . . . . .   7 . . ."..., 48) = 48
745 1674 17:22:40.818094 write(1, "  8 S S S . . . S S S .   8 X . ."..., 48) = 48
746 1674 17:22:40.818466 write(1, "  9 . . . . . . . . . .   9 . . ."..., 48) = 48
747 1674 17:22:40.818878 write(1, "\n", 1) = 1
748 1674 17:22:40.819174 write(1, "=== YOUR TURN ===\n", 18) = 18
749 1674 17:22:40.819451 write(1, "Enter shot coordinates (row colu"..., 37) = 37
750 1674 17:22:40.819732 read(0, "9 7\n", 1024) = 4
751 1674 17:22:42.390015 write(1, "...Sending shot to server...\n", 29) = 29
752 1674 17:22:42.390579 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
753 1674 17:22:42.393227 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
754 1674 17:22:42.395205 write(1, "...Waiting for server to process"..., 41) = 41
755 1674 17:22:42.395600 futex(0x7f7fa8a9c7fc,
      FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, {tv_sec=1765808572,
      tv_nsec=395552001}, FUTEX_BITSET_MATCH_ANY) = 0
756 1674 17:22:42.839394 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
757 1674 17:22:42.847403 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
758 1674 17:22:42.847635 write(1, "*** YOU WON! ***\n", 17) = 17
759 1674 17:22:42.847880 munmap(0x7f7fa8a9c000, 32864) = 0
760 1674 17:22:42.848067 close(3) = 0
761 1674 17:22:42.848226 openat(AT_FDCWD, "players_stats.db", O_RDONLY) = 3
762 1674 17:22:42.848427 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=76004,
      ...}, AT_EMPTY_PATH) = 0
763 1674 17:22:42.848672 read(3,
      "kolko\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 73728) =
      73728
764 1674 17:22:42.848895 read(3,
      "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"...,
      4096) = 2276
765 1674 17:22:42.849159 close(3) = 0
766 1674 17:22:42.849401 write(1, "\n", 1) = 1
767 1674 17:22:42.849610 write(1, "Statistics updated from server\n", 31) = 31
768 1674 17:22:42.849886 write(1, "\n", 1) = 1
769 1674 17:22:42.850108 write(1, "===== MAIN MENU =====\n", 32) = 32
770 1674 17:22:42.850304 write(1, "1. Create new game\n", 19) = 19
771 1674 17:22:42.850485 write(1, "2. Join a game\n", 15) = 15
772 1674 17:22:42.850774 write(1, "3. View my statistics\n", 22) = 22
773 1674 17:22:42.851139 write(1, "4. Exit\n", 8) = 8
774 1674 17:22:42.851349 write(1, "Choose action (1-4): ", 21) = 21

```

[illegible]

Листинг 16: *Strace логи клиента*