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 memory map. 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 (memory-mapped 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 файл 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 акутальности 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 сыграных 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, определённая 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. Одновремено 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 memory 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 memory-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

```c
#ifndef COMMON_H
#define COMMON_H

#include <stdio.h>
#include <string.h>

#include "os.h"
#include "sync.h"

#define BOARD_SIZE 10
#define MAX_GAMES 16
#define MAX_PLAYERS_PER_GAME 2
#define MAX_GAME_NAME 128
#define MAX_LOGIN 64
#define MAX_GAME_ID 32

typedef enum { EMPTY = 0, SHIP = 1, HIT = 2, MISS = 3 } CellState;

typedef enum {
  GAME_WAITING = 0,
  GAME_SETUP = 1,
  GAME_RUNNING = 2,
  GAME_FINISHED = 3
} GameStatus;

typedef struct {
  CellState cells[BOARD_SIZE][BOARD_SIZE];
} Board;

typedef struct {
  char login[MAX_LOGIN];
  int ready;
  Board board;
  Board shots;
} Player;

typedef struct {
  char game_id[MAX_GAME_ID];
  char game_name[MAX_GAME_NAME];
  Player players[MAX_PLAYERS_PER_GAME];
  int player_count;
  int current_turn;
  int status;
  int winner_idx;
  int stats_updated;

  struct {
    int active;
    int row;
    int col;
    int result;
    int processed_by_opponent;
  } last_shots[MAX_PLAYERS_PER_GAME];

  int last_update_version;

  OSSyncMutex game_mutex;
  OSSyncCondVar shot_changed;
} GameState;

typedef struct {
  GameState games[MAX_GAMES];
  int game_count;

  OSSyncMutex state_mutex;
  OSSyncCondVar state_changed;
} ServerState;

typedef struct {
  char login[MAX_LOGIN];
  int wins;
  int losses;
  int games_played;
} PlayerStats;

#endif  // COMMON_H
```

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

```c
#ifndef OS_H
#define OS_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#ifdef _WIN32
#define OS_WINDOWS 1
#define OS_LINUX 0
#else
#define OS_WINDOWS 0
#define OS_LINUX 1
#endif

typedef struct {
  void *addr;
  int fd;
} OSMmapHandle;

```

```c
OSMmapHandle mmap_create(const char *filename, size_t size);

OSMmapHandle mmap_open(const char *filename, size_t size);

void mmap_close(OSMmapHandle handle, size_t size);

void mmap_write(void *addr, const void *data, size_t size);

void mmap_read(void *addr, void *data, size_t size);

void os_file_unlink(const char *filename);

void init_random(void);

int os_get_pid(void);

void os_usleep(unsigned int microseconds);

typedef void (*OSSignalHandler)(int);

void os_signal_register_int(OSSignalHandler handler);

#endif  // OS_H
```

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

```c
#ifndef SYNC_H
#define SYNC_H

#include <pthread.h>

#ifdef _WIN32
#else
typedef pthread_mutex_t OSSyncMutex;
typedef pthread_cond_t OSSyncCondVar;
#endif

int os_mutex_init(OSSyncMutex *mutex);

void os_mutex_destroy(OSSyncMutex *mutex);

void os_mutex_lock(OSSyncMutex *mutex);

void os_mutex_unlock(OSSyncMutex *mutex);

int os_condvar_init(OSSyncCondVar *cond);

void os_condvar_destroy(OSSyncCondVar *cond);

```

```c
int os_condvar_timedwait(OSSyncCondVar *cond, OSSyncMutex *mutex,
                         int timeout_ms);

void os_condvar_broadcast(OSSyncCondVar *cond);

void os_condvar_signal(OSSyncCondVar *cond);

#define OS_WAIT_SUCCESS 0
#define OS_WAIT_TIMEOUT 1
#define OS_WAIT_ERROR -1

#endif  // SYNC_H
```

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

```c
#ifndef MUTEX_INIT_H
#define MUTEX_INIT_H

#include "common.h"

int init_server_mutexes(ServerState *state);

int init_game_mutexes(GameState *game);

void cleanup_server_mutexes(ServerState *state);

#endif
```

Листинг 4: *Объявление функций инициализации примитивов синхронизации*

```c
#ifndef STATS_H
#define STATS_H

#include "common.h"

#define STATS_FILE "players_stats.db"
#define MAX_PLAYERS_STATS 1000

int stats_init();

int load_player_stats(const char *login, PlayerStats *stats);

int save_player_stats(PlayerStats *stats);

int update_player_stats(const char *login, int is_win, int is_loss);

void display_player_stats(const char *login);

```

```c
#endif  // STATS_H
```
Листинг 5: *Заголовочный файл отслеживания статистики*

```c
#ifndef SERVER_H
#define SERVER_H

#include "common.h"

#define STATE_FILE "server_state.mmap"

int server_init(ServerState **state, OSMmapHandle state_handle);

void server_cleanup(ServerState *state, OSMmapHandle state_handle);

int create_game(ServerState *state, const char *game_name,
                const char *player_login, char *out_game_id);

int join_game(ServerState *state, const char *game_id,
              const char *player_login);

GameState *get_game(ServerState *state, const char *game_id);

void list_games(ServerState *state);

int check_game_ready(GameState *game);

void finish_game(GameState *game, int winner_idx);

int process_server_shot(GameState *game, int shooter_idx, int row, int col);

#endif  // SERVER_H
```
Листинг 6: *Заголовочный файл серверных функций*

```c
#ifndef GAME_LOGIC_H
#define GAME_LOGIC_H

#include "common.h"

#define MAX_ATTEMPTS 100

void place_ship(Board *board, int row, int col, int length, int horizontal);

int is_valid_placement(Board *board, int row, int col, int length,
                       int horizontal);

void randomize_board(Board *board);

void init_board(Board *board);

int process_shot(Board *target_board, int row, int col);

int is_ship_sunk(Board *board, int row, int col);

int all_ships_sunk(Board *board);

char cell_to_char(CellState state, int reveal_ships);

#endif  // GAME_LOGIC_H
```

Листинг 7: *Заголовочный файл игровой логики*

```c
#include "os.h"

#if OS_LINUX

#include <fcntl.h>
#include <signal.h>
#include <sys/mman.h>
#include <sys/stat.h>
#include <time.h>

OSMmapHandle mmap_create(const char *filename, size_t size) {
  OSMmapHandle handle = {NULL, -1};
  unlink(filename);

  int fd = open(filename, O_CREAT | O_RDWR | O_TRUNC, 0666);
  if (fd < 0) {
    perror("open");
    return handle;
  }

  if (lseek(fd, size - 1, SEEK_SET) == -1) {
    perror("lseek");
    close(fd);
    return handle;
  }

  if (write(fd, "", 1) != 1) {
    perror("write");
    close(fd);
    return handle;
  }

  void *addr = mmap(NULL, size, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
  if (addr == MAP_FAILED) {
    perror("mmap");
    close(fd);
    return handle;
  }

  memset(addr, 0, size);
  handle.addr = addr;
  handle.fd = fd;

  return handle;
}

OSMmapHandle mmap_open(const char *filename, size_t size) {
  OSMmapHandle handle = {NULL, -1};

  int fd = open(filename, O_RDWR, 0666);
  if (fd < 0) {
    perror("open");
    return handle;
  }

  void *addr = mmap(NULL, size, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
  if (addr == MAP_FAILED) {
    perror("mmap");
    close(fd);
    return handle;
  }

  handle.addr = addr;
  handle.fd = fd;

  return handle;
}

void mmap_close(OSMmapHandle handle, size_t size) {
  if (handle.addr != NULL && handle.addr != MAP_FAILED) {
    munmap(handle.addr, size);
  }

  if (handle.fd >= 0) {
    close(handle.fd);
  }
}

void mmap_write(void *addr, const void *data, size_t size) {
  if (addr && data) {
    memcpy(addr, data, size);
    msync(addr, size, MS_SYNC);
  }
}

void mmap_read(void *addr, void *data, size_t size) {
  if (addr && data) {
    memcpy(data, addr, size);
  }
}

void os_file_unlink(const char *filename) { unlink(filename); }

void init_random(void) { srand((unsigned int)time(NULL) ^ getpid()); }

int os_get_pid(void) { return getpid(); }

void os_usleep(unsigned int microseconds) { usleep(microseconds); }

void os_signal_register_int(OSSignalHandler handler) {
  signal(SIGINT, handler);
}

#endif  // OS_LINUX
```

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

```c
#include "sync.h"

#include <stdio.h>

#include "sync_init.h"

#ifdef _WIN32
#else
#include <errno.h>
#include <pthread.h>
#include <time.h>
#endif

int os_mutex_init(OSSyncMutex *mutex) {
  pthread_mutexattr_t attr;

  if (pthread_mutexattr_init(&attr) != 0) {
    perror("pthread_mutexattr_init");
    return -1;
  }

  if (pthread_mutexattr_setpshared(&attr, PTHREAD_PROCESS_SHARED) != 0) {
    perror("pthread_mutexattr_setpshared");
    pthread_mutexattr_destroy(&attr);
    return -1;
  }

  if (pthread_mutexattr_settype(&attr, PTHREAD_MUTEX_RECURSIVE) != 0) {
    perror("pthread_mutexattr_settype");
    pthread_mutexattr_destroy(&attr);
    return -1;
  }

  int result = pthread_mutex_init(mutex, &attr);
  pthread_mutexattr_destroy(&attr);
  return result;
}

void os_mutex_destroy(OSSyncMutex *mutex) {
  if (mutex) {
    pthread_mutex_destroy(mutex);
  }
}

void os_mutex_lock(OSSyncMutex *mutex) {
  if (mutex) {
    pthread_mutex_lock(mutex);
  }
}

void os_mutex_unlock(OSSyncMutex *mutex) {
  if (mutex) {
    pthread_mutex_unlock(mutex);
  }
}

int os_condvar_init(OSSyncCondVar *cond) {
  pthread_condattr_t attr;

  if (pthread_condattr_init(&attr) != 0) {
    perror("pthread_condattr_init");
    return -1;
  }

  if (pthread_condattr_setpshared(&attr, PTHREAD_PROCESS_SHARED) != 0) {
    perror("pthread_condattr_setpshared");
    pthread_condattr_destroy(&attr);
    return -1;
  }

  int result = pthread_cond_init(cond, &attr);
  pthread_condattr_destroy(&attr);
  return result;
}

void os_condvar_destroy(OSSyncCondVar *cond) {
  if (cond) {
    pthread_cond_destroy(cond);
  }
}
```

```c

int os_condvar_timedwait(OSSyncCondVar *cond, OSSyncMutex *mutex,
                         int timeout_ms) {
  if (!cond || !mutex) {
    return OS_WAIT_ERROR;
  }

  struct timespec timeout;
  if (clock_gettime(CLOCK_REALTIME, &timeout) != 0) {
    perror("clock_gettime");
    return OS_WAIT_ERROR;
  }

  long ns = (long)timeout.tv_nsec + ((long)timeout_ms * 1000000);
  timeout.tv_sec += ns / 1000000000;
  timeout.tv_nsec = ns % 1000000000;

  int result = pthread_cond_timedwait(cond, mutex, &timeout);

  if (result == ETIMEDOUT) {
    return OS_WAIT_TIMEOUT;
  } else if (result == 0) {
    return OS_WAIT_SUCCESS;
  } else {
    return OS_WAIT_ERROR;
  }
}

void os_condvar_broadcast(OSSyncCondVar *cond) {
  if (cond) {
    pthread_cond_broadcast(cond);
  }
}

void os_condvar_signal(OSSyncCondVar *cond) {
  if (cond) {
    pthread_cond_signal(cond);
  }
}

int init_server_mutexes(ServerState *state) {
  if (!state) return 0;

  printf("Initialized state_mutex\n");
  if (os_mutex_init(&state->state_mutex) != 0) {
    return 0;
  }

  printf("Initialized state_changed condition variable\n");
  if (os_condvar_init(&state->state_changed) != 0) {
    os_mutex_destroy(&state->state_mutex);
    return 0;
  }

  return 1;
}

int init_game_mutexes(GameState *game) {
  if (!game) return 0;

  if (os_mutex_init(&game->game_mutex) != 0) {
    return 0;
  }

  if (os_condvar_init(&game->shot_changed) != 0) {
    os_mutex_destroy(&game->game_mutex);
    return 0;
  }

  return 1;
}

void cleanup_server_mutexes(ServerState *state) {
  if (!state) return;

  for (int i = 0; i < state->game_count; i++) {
    os_mutex_destroy(&state->games[i].game_mutex);
    os_condvar_destroy(&state->games[i].shot_changed);
  }

  os_mutex_destroy(&state->state_mutex);
  os_condvar_destroy(&state->state_changed);

  printf("Synchronization primitives cleaned up\n");
}
```

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

```
#include "stats.h"

typedef struct {
  PlayerStats players[MAX_PLAYERS_STATS];
  int count;
} StatsDatabase;

static StatsDatabase db;
static int db_initialized = 0;

int stats_init() {
  memset(&db, 0, sizeof(StatsDatabase));

```

```c
  FILE *f = fopen(STATS_FILE, "rb");
  if (f) {
    fread(&db, sizeof(StatsDatabase), 1, f);
    fclose(f);
  }

  db_initialized = 1;
  return 1;
}

static int find_or_create_player(const char *login) {
  for (int i = 0; i < db.count; i++) {
    if (strcmp(db.players[i].login, login) == 0) {
      return i;
    }
  }

  if (db.count >= MAX_PLAYERS_STATS) {
    fprintf(stderr, "Error: statistics database is full\n");
    return -1;
  }

  int idx = db.count;
  memset(&db.players[idx], 0, sizeof(PlayerStats));
  strcpy(db.players[idx].login, login);
  db.count++;
  return idx;
}

int load_player_stats(const char *login, PlayerStats *stats) {
  if (!db_initialized) stats_init();

  int idx = find_or_create_player(login);
  if (idx < 0) return 0;

  *stats = db.players[idx];
  return 1;
}

int save_player_stats(PlayerStats *stats) {
  if (!db_initialized) stats_init();

  int idx = find_or_create_player(stats->login);
  if (idx < 0) return 0;

  db.players[idx] = *stats;

  FILE *f = fopen(STATS_FILE, "wb");
  if (!f) {
    perror("fopen");
    return 0;
  }

  fwrite(&db, sizeof(StatsDatabase), 1, f);
  fclose(f);
  return 1;
}

int update_player_stats(const char *login, int is_win, int is_loss) {
  if (!db_initialized) stats_init();

  PlayerStats stats;
  load_player_stats(login, &stats);

  if (is_win) {
    stats.wins++;
    stats.games_played++;
  } else if (is_loss) {
    stats.losses++;
    stats.games_played++;
  }

  return save_player_stats(&stats);
}

void display_player_stats(const char *login) {
  if (!db_initialized) stats_init();

  PlayerStats stats;
  load_player_stats(login, &stats);

  printf("\n========== PLAYER STATISTICS ==========\n");
  printf("Login: %s\n", stats.login);
  printf("Wins: %d\n", stats.wins);
  printf("Loses: %d\n", stats.losses);
  printf("Total games: %d\n", stats.games_played);

  if (stats.games_played > 0) {
    double win_rate = (double)stats.wins / stats.games_played * 100;
    printf("Win rate: %.1f%%\n", win_rate);
  }
}
```

Листинг 10: *Реализация отслеживания статистики*

```c
#include "server.h"

#include "game_logic.h"
#include "stats.h"
#include "sync_init.h"

```

```c
int server_init(ServerState **state, OSMmapHandle state_handle) {
  os_file_unlink(STATE_FILE);
  state_handle = mmap_create(STATE_FILE, sizeof(ServerState));
  *state = (ServerState *)state_handle.addr;

  if (*state == NULL) {
    fprintf(stderr, "Error: failed to create state file\n");
    return 0;
  }

  (*state)->game_count = 0;
  printf("Server initialized\n");
  return 1;
}

void server_cleanup(ServerState *state, OSMmapHandle handle) {
  mmap_close(handle, sizeof(ServerState));
}

int create_game(ServerState *state, const char *game_name,
                const char *player_login, char *out_game_id) {
  os_mutex_lock(&state->state_mutex);

  if (state->game_count >= MAX_GAMES) {
    fprintf(stderr, "Error: maximum number of games reached\n");
    os_mutex_unlock(&state->state_mutex);
    return 0;
  }

  GameState *game = &state->games[state->game_count];
  snprintf(out_game_id, MAX_GAME_ID, "%d", state->game_count);
  strcpy(game->game_id, out_game_id);
  strcpy(game->game_name, game_name);

  strcpy(game->players[0].login, player_login);
  game->players[0].ready = 0;
  init_board(&game->players[0].board);
  init_board(&game->players[0].shots);

  game->player_count = 1;
  game->current_turn = 0;
  game->status = GAME_WAITING;
  game->winner_idx = -1;
  game->stats_updated = 0;

  for (int i = 0; i < 2; i++) {
    game->last_shots[i].result = -1;
    game->last_shots[i].processed_by_opponent = 0;
  }

  if (!init_game_mutexes(game)) {
    fprintf(stderr, "Error initializing game mutexes\n");
    os_mutex_unlock(&state->state_mutex);
    return 0;
  }

  game->last_update_version = 0;
  state->game_count++;
  os_condvar_broadcast(&state->state_changed);
  os_mutex_unlock(&state->state_mutex);

  printf("Game '%s' created (ID: %s)\n", game_name, out_game_id);
  return 1;
}

int join_game(ServerState *state, const char *game_id,
              const char *player_login) {
  os_mutex_lock(&state->state_mutex);

  for (int i = 0; i < state->game_count; i++) {
    if (strcmp(state->games[i].game_id, game_id) == 0) {
      GameState *game = &state->games[i];

      if (game->player_count >= MAX_PLAYERS_PER_GAME) {
        fprintf(stderr, "Error: game is full\n");
        os_mutex_unlock(&state->state_mutex);
        return 0;
      }

      if (game->status != GAME_WAITING) {
        fprintf(stderr, "Error: game already started\n");
        os_mutex_unlock(&state->state_mutex);
        return 0;
      }

      os_mutex_lock(&game->game_mutex);

      if (game->player_count >= MAX_PLAYERS_PER_GAME) {
        fprintf(stderr, "Error: game is full\n");
        os_mutex_unlock(&game->game_mutex);
        os_mutex_unlock(&state->state_mutex);
        return 0;
      }

      strcpy(game->players[1].login, player_login);
      game->players[1].ready = 0;
      init_board(&game->players[1].board);
      init_board(&game->players[1].shots);
      game->player_count++;
      game->status = GAME_SETUP;

      os_condvar_broadcast(&game->shot_changed);
      os_condvar_broadcast(&state->state_changed);
      os_mutex_unlock(&game->game_mutex);
      os_mutex_unlock(&state->state_mutex);

      printf("Player '%s' joined the game\n", player_login);
      return 1;
    }
  }

  os_mutex_unlock(&state->state_mutex);
  fprintf(stderr, "Error: game not found\n");
  return 0;
}

GameState *get_game(ServerState *state, const char *game_id) {
  for (int i = 0; i < state->game_count; i++) {
    if (strcmp(state->games[i].game_id, game_id) == 0) {
      return &state->games[i];
    }
  }
  return NULL;
}

void list_games(ServerState *state) {
  printf("\n========== AVAILABLE GAMES ==========\n");

  if (state->game_count == 0) {
    printf("No available games\n");
    return;
  }

  for (int i = 0; i < state->game_count; i++) {
    GameState *game = &state->games[i];

    if (game->status == GAME_FINISHED) {
      continue;
    }

    printf("%d. [%s] %s - %d/%d players (status: ", i + 1, game->game_id,
           game->game_name, game->player_count, MAX_PLAYERS_PER_GAME);

    switch (game->status) {
      case GAME_WAITING:
        printf("waiting");
        break;
      case GAME_SETUP:
        printf("setup");
        break;
      case GAME_RUNNING:
        printf("running");
        break;
      case GAME_FINISHED:
        printf("finished");
        break;
      default:
        printf("?");
        break;
    }
    printf(")\n");
  }
}

int check_game_ready(GameState *game) {
  if (game->player_count != 2) {
    return 0;
  }

  if (game->players[0].ready && game->players[1].ready) {
    game->status = GAME_RUNNING;
    return 1;
  }

  return 0;
}

void finish_game(GameState *game, int winner_idx) {
  if (game->status != GAME_RUNNING) {
    return;
  }

  game->status = GAME_FINISHED;
  game->winner_idx = winner_idx;

  if (winner_idx >= 0 && winner_idx < 2) {
    update_player_stats(game->players[winner_idx].login, 1, 0);
    int loser_idx = 1 - winner_idx;
    update_player_stats(game->players[loser_idx].login, 0, 1);
    printf("Game finished! Winner: %s\n", game->players[winner_idx].login);
  }
}

int process_server_shot(GameState *game, int shooter_idx, int row, int col) {
  if (game->status != GAME_RUNNING) {
    return -1;
  }

  int opponent_idx = 1 - shooter_idx;

  if (row < 0 || row >= BOARD_SIZE || col < 0 || col >= BOARD_SIZE) {
    return -1;
  }

```

```c
  Board *shooter_shots = &game->players[shooter_idx].shots;

  if (shooter_shots->cells[row][col] != EMPTY) {
    return -1;
  }

  Board *opponent_board = &game->players[opponent_idx].board;
  int result = process_shot(opponent_board, row, col);

  if (result == -1) {
    return -1;
  }

  if (result == 0) {
    shooter_shots->cells[row][col] = MISS;
  } else {
    shooter_shots->cells[row][col] = HIT;
  }

  game->players[opponent_idx].board = *opponent_board;
  game->players[shooter_idx].shots = *shooter_shots;

  printf("[SHOT] Player %d shoots [%d,%d] -> ", shooter_idx, row, col);

  if (result == 0) {
    game->current_turn = opponent_idx;
    printf("MISS. Turn goes to player %d\n", opponent_idx);
  } else if (result == 1) {
    game->current_turn = shooter_idx;
    printf("HIT! Turn stays with player %d\n", shooter_idx);
  } else if (result == 2) {
    game->current_turn = shooter_idx;
    printf("SUNK! Turn stays with player %d\n", shooter_idx);
  }

  if (all_ships_sunk(opponent_board)) {
    game->status = GAME_FINISHED;
    game->winner_idx = shooter_idx;
    printf("[VICTORY] Player %d won!\n", shooter_idx);
    return 3;
  }

  return result;
}
```

Листинг 11: *Реализация серверных функций*

```c
#include "game_logic.h"

void init_board(Board *board) {
  for (int i = 0; i < BOARD_SIZE; i++) {
    for (int j = 0; j < BOARD_SIZE; j++) {
      board->cells[i][j] = EMPTY;
    }
  }
}

int is_valid_placement(Board *board, int row, int col, int length,
                       int horizontal) {
  if (horizontal) {
    if (col + length > BOARD_SIZE) return 0;
  } else {
    if (row + length > BOARD_SIZE) return 0;
  }

  for (int i = 0; i < length; i++) {
    int r = horizontal ? row : row + i;
    int c = horizontal ? col + i : col;

    if (board->cells[r][c] != EMPTY) {
      return 0;
    }
  }

  if (horizontal) {
    if (col > 0) {
      for (int r = row - 1; r <= row + 1; r++) {
        if (r >= 0 && r < BOARD_SIZE) {
          if (board->cells[r][col - 1] != EMPTY) return 0;
        }
      }
    }

    if (col + length < BOARD_SIZE) {
      for (int r = row - 1; r <= row + 1; r++) {
        if (r >= 0 && r < BOARD_SIZE) {
          if (board->cells[r][col + length] != EMPTY) return 0;
        }
      }
    }

    if (row > 0) {
      for (int c = col - 1; c <= col + length; c++) {
        if (c >= 0 && c < BOARD_SIZE) {
          if (board->cells[row - 1][c] != EMPTY) return 0;
        }
      }
    }
    if (row < BOARD_SIZE - 1) {
      for (int c = col - 1; c <= col + length; c++) {
        if (c >= 0 && c < BOARD_SIZE) {
          if (board->cells[row + 1][c] != EMPTY) return 0;
        }
      }
    }
  } else {
    if (row > 0) {
      for (int c = col - 1; c <= col + 1; c++) {
        if (c >= 0 && c < BOARD_SIZE) {
          if (board->cells[row - 1][c] != EMPTY) return 0;
        }
      }
    }

    if (row + length < BOARD_SIZE) {
      for (int c = col - 1; c <= col + 1; c++) {
        if (c >= 0 && c < BOARD_SIZE) {
          if (board->cells[row + length][c] != EMPTY) return 0;
        }
      }
    }

    if (col > 0) {
      for (int r = row - 1; r <= row + length; r++) {
        if (r >= 0 && r < BOARD_SIZE) {
          if (board->cells[r][col - 1] != EMPTY) return 0;
        }
      }
    }
    if (col < BOARD_SIZE - 1) {
      for (int r = row - 1; r <= row + length; r++) {
        if (r >= 0 && r < BOARD_SIZE) {
          if (board->cells[r][col + 1] != EMPTY) return 0;
        }
      }
    }
  }

  return 1;
}

void place_ship(Board *board, int row, int col, int length, int horizontal) {
  if (!is_valid_placement(board, row, col, length, horizontal)) {
    return;
  }

  for (int i = 0; i < length; i++) {
    int r = horizontal ? row : row + i;
    int c = horizontal ? col + i : col;
    board->cells[r][c] = SHIP;
  }
}
```

```c

void randomize_board(Board *board) {
  init_board(board);

  int ship_lengths[] = {4, 3, 3, 2, 2, 2, 1, 1, 1, 1};
  int num_ships = 10;

  for (int ship_idx = 0; ship_idx < num_ships; ship_idx++) {
    int length = ship_lengths[ship_idx];
    int placed = 0;
    int attempts = 0;

    while (!placed && attempts < MAX_ATTEMPTS) {
      int row = rand() % BOARD_SIZE;
      int col = rand() % BOARD_SIZE;
      int horizontal = rand() % 2;

      if (is_valid_placement(board, row, col, length, horizontal)) {
        place_ship(board, row, col, length, horizontal);
        placed = 1;
      }

      attempts++;
    }

    if (!placed) {
      ship_idx--;
      if (ship_idx < 0) ship_idx = 0;
      init_board(board);
    }
  }
}

int is_ship_sunk(Board *board, int row, int col) {
  if (board->cells[row][col] != HIT) {
    return 0;
  }

  int col_start = col;
  int col_end = col;

  while (col_start > 0 && (board->cells[row][col_start - 1] == SHIP ||
                           board->cells[row][col_start - 1] == HIT)) {
    col_start--;
  }

  while (col_end < BOARD_SIZE - 1 && (board->cells[row][col_end + 1] == SHIP ||
                                      board->cells[row][col_end + 1] == HIT)) {
    col_end++;
  }

  int horizontal_intact = 0;
  for (int c = col_start; c <= col_end; c++) {
    if (board->cells[row][c] == SHIP) {
      horizontal_intact = 1;
      break;
    }
  }

  if (!horizontal_intact && col_start < col_end) {
    return 1;
  }

  int row_start = row;
  int row_end = row;

  while (row_start > 0 && (board->cells[row_start - 1][col] == SHIP ||
                           board->cells[row_start - 1][col] == HIT)) {
    row_start--;
  }

  while (row_end < BOARD_SIZE - 1 && (board->cells[row_end + 1][col] == SHIP ||
                                      board->cells[row_end + 1][col] == HIT)) {
    row_end++;
  }

  int vertical_intact = 0;
  for (int r = row_start; r <= row_end; r++) {
    if (board->cells[r][col] == SHIP) {
      vertical_intact = 1;
      break;
    }
  }

  if (!vertical_intact && row_start < row_end) {
    return 1;
  }

  if (!horizontal_intact && col_start == col_end && !vertical_intact &&
      row_start == row_end) {
    return 1;
  }

  return 0;
}

int process_shot(Board *target_board, int row, int col) {
  CellState cell = target_board->cells[row][col];

  if (cell == EMPTY) {
    target_board->cells[row][col] = MISS;
    return 0;
  } else if (cell == SHIP) {
    target_board->cells[row][col] = HIT;
    if (is_ship_sunk(target_board, row, col)) {
      return 2;
    }
    return 1;
  } else if (cell == HIT || cell == MISS) {
    return -1;
  }

  return 0;
}

int all_ships_sunk(Board *board) {
  for (int i = 0; i < BOARD_SIZE; i++) {
    for (int j = 0; j < BOARD_SIZE; j++) {
      if (board->cells[i][j] == SHIP) {
        return 0;
      }
    }
  }
  return 1;
}

char cell_to_char(CellState state, int reveal_ships) {
  switch (state) {
    case EMPTY:
      return '.';
    case SHIP:
      return reveal_ships ? 'S' : '.';
    case HIT:
      return 'X';
    case MISS:
      return 'O';
    default:
      return '?';
  }
}
```

Листинг 12: *Реализация игровой логики*

```c
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include "common.h"
#include "game_logic.h"
#include "server.h"
```

```c
#include "stats.h"
#include "sync_init.h"

static int keep_running = 1;

void signal_handler(int sig) {
  (void)sig;
  keep_running = 0;
}

int main(int argc, char *argv[]) {
  (void)argc;
  (void)argv;


   printf("
  printf(" BATTLESHIP - SERVER \n");

   printf("

  os_signal_register_int(signal_handler);

  OSMmapHandle state_handle;
  ServerState *state = NULL;

  if (!server_init(&state, state_handle)) {
    fprintf(stderr, "Error initializing server\n");
    return 1;
  }

  printf("Initializing synchronization primitives...\n");
  if (!init_server_mutexes(state)) {
    fprintf(stderr, "Error initializing mutexes\n");
    server_cleanup(state, state_handle);
    return 1;
  }

  printf("Synchronization primitives initialized\n\n");

  stats_init();
  printf("Statistics system initialized\n\n");

  printf("Server is running and waiting for players...\n");
  printf("Clients can connect\n\n");

  int last_game_count = 0;

  while (keep_running) {
    os_usleep(500000);

    OSMmapHandle handle = mmap_open(STATE_FILE, sizeof(ServerState));
    if (handle.addr == NULL) continue;

    mmap_read(handle.addr, state, sizeof(ServerState));

    os_mutex_lock(&state->state_mutex);

    for (int i = 0; i < state->game_count; i++) {
      GameState *game = &state->games[i];
      os_mutex_lock(&game->game_mutex);

      for (int shooter_idx = 0; shooter_idx < 2; shooter_idx++) {
        if (game->last_shots[shooter_idx].active &&
            game->last_shots[shooter_idx].result == -2) {
          int row = game->last_shots[shooter_idx].row;
          int col = game->last_shots[shooter_idx].col;

          printf("\n[GAME: %s] Processing shot from player %d\n",
                 game->game_name, shooter_idx);

          if (row < 0 || row >= BOARD_SIZE || col < 0 || col >= BOARD_SIZE) {
            printf("Error: coordinates out of bounds\n");
            game->last_shots[shooter_idx].result = -1;
          } else {
            int result = process_server_shot(game, shooter_idx, row, col);
            game->last_shots[shooter_idx].result = result;

            switch (result) {
              case 0:
                printf("MISS [%d,%d]\n", row, col);
                break;
              case 1:
                printf("HIT [%d,%d]\n", row, col);
                break;
              case 2:
                printf("SUNK [%d,%d]\n", row, col);
                break;
              case 3:
                printf("*** VICTORY! Player %d won!\n", shooter_idx);
                game->status = GAME_FINISHED;
                game->winner_idx = shooter_idx;
                break;
              case -1:
                printf("ERROR during processing\n");
                break;
            }

            game->last_update_version++;
            os_condvar_broadcast(&game->shot_changed);
            printf("Result sent (version %d)\n", game->last_update_version);
          }
        }
```

```c
      }

      if (game->status == GAME_FINISHED && game->winner_idx >= 0 &&
          !game->stats_updated) {
        printf("\n[GAME %s] Finished!\n", game->game_name);
        printf(" Winner: %s\n", game->players[game->winner_idx].login);
        printf(" Loser: %s\n", game->players[1 - game->winner_idx].login);

        update_player_stats(game->players[game->winner_idx].login, 1, 0);
        int loser_idx = 1 - game->winner_idx;
        update_player_stats(game->players[loser_idx].login, 0, 1);

        game->stats_updated = 1;
        game->last_shots[0].active = 0;
        game->last_shots[1].active = 0;
        os_condvar_broadcast(&game->shot_changed);
      }

      os_mutex_unlock(&game->game_mutex);
    }

    os_mutex_unlock(&state->state_mutex);
    mmap_close(handle, sizeof(ServerState));

    if (state->game_count != last_game_count) {
      printf("\n[SERVER] Active games: %d/%d\n", state->game_count, MAX_GAMES);
      last_game_count = state->game_count;
    }
  }

  printf("\n[SERVER] Shutting down...\n");

  for (int i = 0; i < state->game_count; i++) {
    GameState *game = &state->games[i];

    os_mutex_lock(&game->game_mutex);
    game->status = GAME_FINISHED;
    game->winner_idx = -1;
    os_condvar_broadcast(&game->shot_changed);
    os_mutex_unlock(&game->game_mutex);
  }

  os_usleep(5000000);
  cleanup_server_mutexes(state);
  server_cleanup(state, state_handle);
  printf("Server stopped\n");

  return 0;
}
```

Листинг 13: *Реализация сервера*

```c
#include <errno.h>
#include <pthread.h>
#include <time.h>

#include "common.h"
#include "game_logic.h"
#include "server.h"
#include "stats.h"

#define SHOT_TIMEOUT 300

static char player_login[MAX_LOGIN];
static char current_game_id[MAX_GAME_ID];
static int player_index = -1;

static ServerState *get_server_state(OSMmapHandle *handle) {
  *handle = mmap_open("server_state.mmap", sizeof(ServerState));
  return (ServerState *)handle->addr;
}

static void clear_input_buffer(void) {
  int c;
  while ((c = getchar()) != '\n' && c != EOF) {
  }
}

static int login_menu() {
  printf("\n==========================================\n");
  printf("=  BATTLESHIP - CLIENT                  =\n");
  printf("==========================================\n\n");

  printf("Enter your login (up to 63 characters): ");
  if (fgets(player_login, sizeof(player_login), stdin) == NULL) {
    return 0;
  }
  player_login[strcspn(player_login, "\n")] = 0;

  if (strlen(player_login) == 0) {
    printf("Error: login cannot be empty\n");
    return 0;
  }

  printf("Welcome, %s!\n", player_login);
  stats_init();

  return 1;
}

static int main_menu() {
  printf("\n========== MAIN MENU ==========\n");
  printf("1. Create new game\n");
  printf("2. Join a game\n");
  printf("3. View my statistics\n");
  printf("4. Exit\n");
  printf("Choose action (1-4): ");

  char choice[10];
  if (fgets(choice, sizeof(choice), stdin) == NULL) {
    return 4;
  }

  return atoi(choice);
}

static int create_game_dialog() {
  char game_name[MAX_GAME_NAME];

  printf("\nEnter name of new game: ");
  if (fgets(game_name, sizeof(game_name), stdin) == NULL) {
    return 0;
  }
  game_name[strcspn(game_name, "\n")] = 0;

  if (strlen(game_name) == 0) {
    printf("Error: game name cannot be empty\n");
    return 0;
  }

  OSMmapHandle fd;
  ServerState *state = get_server_state(&fd);
  if (state == NULL) {
    fprintf(stderr, "Error: could not connect to server\n");
    return 0;
  }

  if (create_game(state, game_name, player_login, current_game_id)) {
    mmap_close(fd, sizeof(ServerState));
    player_index = 0;
    printf("Game created! ID: %s\n", current_game_id);
    printf("Waiting for second player...\n");
    return 1;
  }

  mmap_close(fd, sizeof(ServerState));
  return 0;
}

static int join_game_dialog() {
  OSMmapHandle fd;
  ServerState *state = get_server_state(&fd);
  if (state == NULL) {
    fprintf(stderr, "Error: could not connect to server\n");
    return 0;
  }

  list_games(state);

  printf("\nEnter game ID to join (or empty to cancel): ");
  char game_id[MAX_GAME_ID];
  if (fgets(game_id, sizeof(game_id), stdin) == NULL) {
    mmap_close(fd, sizeof(ServerState));
    return 0;
  }
  game_id[strcspn(game_id, "\n")] = 0;

  if (strlen(game_id) == 0) {
    mmap_close(fd, sizeof(ServerState));
    return 0;
  }

  if (join_game(state, game_id, player_login)) {
    strcpy(current_game_id, game_id);
    mmap_close(fd, sizeof(ServerState));
    player_index = 1;
    printf("Joined the game!\n");
    return 1;
  }

  mmap_close(fd, sizeof(ServerState));
  return 0;
}

static void setup_ships(Board *board) {
  printf("\n========== SHIP PLACEMENT ==========\n");
  printf("Generating random ship placement...\n");
  printf("- 1 four-deck ship (1x4)\n");
  printf("- 2 three-deck ships (2x3)\n");
  printf("- 3 two-deck ships (3x2)\n");
  printf("- 4 one-deck ships (4x1)\n\n");

  randomize_board(board);

  printf("\n%s\n", "Ships placed randomly!");
  printf("   0 1 2 3 4 5 6 7 8 9\n");
  for (int i = 0; i < BOARD_SIZE; i++) {
    printf(" %d ", i);
    for (int j = 0; j < BOARD_SIZE; j++) {
      printf("%c ", cell_to_char(board->cells[i][j], 1));
    }
    printf("\n");
  }
}

```

```c
static int get_opponent_index() { return (player_index == 0) ? 1 : 0; }

static void display_boards(const Board *my_board, const Board *my_shots) {
  printf("\nYour board:              Your shots:\n");
  printf("   0 1 2 3 4 5 6 7 8 9    0 1 2 3 4 5 6 7 8 9\n");

  for (int i = 0; i < BOARD_SIZE; i++) {
    printf(" %d ", i);
    for (int j = 0; j < BOARD_SIZE; j++) {
      printf("%c ", cell_to_char(my_board->cells[i][j], 1));
    }
    printf("  %d ", i);
    for (int j = 0; j < BOARD_SIZE; j++) {
      printf("%c ", cell_to_char(my_shots->cells[i][j], 0));
    }
    printf("\n");
  }
}

int wait_for_shot_processing(GameState *game, int player_idx, int max_seconds) {
  printf("...Waiting for server to process shot...\n");

  os_mutex_lock(&game->game_mutex);

  while (game->last_shots[player_idx].result == -2) {
    int wait_result = os_condvar_timedwait(
        &game->shot_changed, &game->game_mutex, max_seconds * 1000);

    if (wait_result == OS_WAIT_TIMEOUT) {
      printf("Timeout: server did not respond in %d seconds\n", max_seconds);
      os_mutex_unlock(&game->game_mutex);
      return 0;
    }

    if (wait_result == OS_WAIT_ERROR) {
      printf("Wait error\n");
      os_mutex_unlock(&game->game_mutex);
      return 0;
    }
  }

  os_mutex_unlock(&game->game_mutex);

  return 1;
}

static void play_game() {
  printf("\n========== GAME START ==========\n");
  OSMmapHandle fd;
  ServerState *state = get_server_state(&fd);

  if (state == NULL) {
    fprintf(stderr, "Error: could not connect to server\n");
    return;
  }

  GameState *game = get_game(state, current_game_id);
  if (game == NULL) {
    fprintf(stderr, "Error: game not found\n");
    mmap_close(fd, sizeof(ServerState));
    return;
  }

  int opponent_idx = get_opponent_index();
  int game_over = 0;
  int last_opponent_shot_version = -1;

  while (!game_over) {
    mmap_read(state, state, sizeof(ServerState));
    game = get_game(state, current_game_id);

    if (game == NULL) {
      fprintf(stderr, "Error: game lost\n");
      break;
    }

    if (game->last_shots[opponent_idx].result == 3) {
      printf("\n~~~ YOU LOST! ~~~\n");
      game->status = GAME_FINISHED;
      game->winner_idx = opponent_idx;
      game_over = 1;
      break;
    }

    Board my_board = game->players[player_index].board;
    Board my_shots = game->players[player_index].shots;

    printf("\n========== CURRENT STATUS ==========\n");

    display_boards(&my_board, &my_shots);

    if (game->current_turn == player_index) {
      printf("\n=== YOUR TURN ===\n");
      printf("Enter shot coordinates (row column): ");

      int row, col;
      if (scanf("%d %d", &row, &col) != 2) {
        clear_input_buffer();
        printf("Error: enter two numbers!\n");
        continue;
      }

```

```
      clear_input_buffer();

      if (row < 0 || row >= BOARD_SIZE || col < 0 || col >= BOARD_SIZE) {
        printf("Error: coordinates out of bounds (0-%d)\n", BOARD_SIZE - 1);
        continue;
      }

      if (my_shots.cells[row][col] != EMPTY) {
        printf("Error: you already shot here!\n");
        continue;
      }

      printf("...Sending shot to server...\n");

      os_mutex_lock(&game->game_mutex);

      game->last_shots[player_index].active = 1;
      game->last_shots[player_index].row = row;
      game->last_shots[player_index].col = col;
      game->last_shots[player_index].result = -2;
      game->last_shots[player_index].processed_by_opponent = 0;
      game->last_update_version++;

      for (int i = 0; i < state->game_count; i++) {
        if (strcmp(state->games[i].game_id, current_game_id) == 0) {
          state->games[i] = *game;
          mmap_write(&state->games[i], &state->games[i], sizeof(GameState));
          break;
        }
      }

      mmap_write(state, state, sizeof(ServerState));

      os_mutex_unlock(&game->game_mutex);

      if (!wait_for_shot_processing(game, player_index, 10)) {
        printf("Error: server did not process shot\n");
        continue;
      }

      mmap_read(state, state, sizeof(ServerState));
      game = get_game(state, current_game_id);
      if (game == NULL) break;

      int result = game->last_shots[player_index].result;

      if (result == -1) {
        printf("Error: invalid shot\n");
      } else if (result == 0) {
        printf("MISS! Turn goes to opponent.\n");
      } else if (result == 1) {
        printf("HIT! You shoot again!\n");
      } else if (result == 2) {
        printf("SUNK! You shoot again!\n");
      } else if (result == 3) {
        printf("*** YOU WON! ***\n");
        game_over = 1;
        break;
      }

      os_mutex_lock(&game->game_mutex);
      game->last_shots[player_index].active = 0;
      os_mutex_unlock(&game->game_mutex);

      for (int i = 0; i < state->game_count; i++) {
        if (strcmp(state->games[i].game_id, current_game_id) == 0) {
          state->games[i] = *game;
          mmap_write(&state->games[i], &state->games[i], sizeof(GameState));
          break;
        }
      }
      mmap_write(state, state, sizeof(ServerState));

    } else {
      printf("\n=== WAITING FOR OPPONENT'S TURN ===\n");

      int opponent_made_move = 0;

      os_mutex_lock(&game->game_mutex);

      while (game->current_turn == opponent_idx &&
             game->status != GAME_FINISHED && !opponent_made_move) {
        int wait_result = os_condvar_timedwait(&game->shot_changed,
                                                &game->game_mutex, 120000);

        if (wait_result == OS_WAIT_TIMEOUT) {
          printf("Opponent is taking too long\n");
          os_mutex_unlock(&game->game_mutex);
          break;
        }

        if (wait_result == OS_WAIT_ERROR) {
          os_mutex_unlock(&game->game_mutex);
          break;
        }

        if (game->last_update_version != last_opponent_shot_version) {
          opponent_made_move = 1;
        }
      }

      os_mutex_unlock(&game->game_mutex);
```

```

    mmap_read(state, state, sizeof(ServerState));
    game = get_game(state, current_game_id);
    if (game == NULL) break;

    my_board = game->players[player_index].board;
    my_shots = game->players[player_index].shots;

    if (game->last_shots[opponent_idx].result >= -1 &&
        last_opponent_shot_version != game->last_update_version) {
      last_opponent_shot_version = game->last_update_version;

      printf("\n*** OPPONENT SHOT! ***\n");
      int row = game->last_shots[opponent_idx].row;
      int col = game->last_shots[opponent_idx].col;
      int result = game->last_shots[opponent_idx].result;

      printf("Opponent shot [%d,%d]: ", row, col);

      if (result == 0) {
        printf("MISS\n");
        my_board.cells[row][col] = MISS;
      } else if (result == 1) {
        printf("HIT your cell!\n");
        my_board.cells[row][col] = HIT;
      } else if (result == 2) {
        printf("SUNK your ship!\n");
        my_board.cells[row][col] = HIT;
      } else if (result == 3) {
        printf("~~~YOU LOST!~~~\n");
      }

      game->last_shots[opponent_idx].processed_by_opponent = 1;

      for (int i = 0; i < state->game_count; i++) {
        if (strcmp(state->games[i].game_id, current_game_id) == 0) {
          state->games[i] = *game;
          mmap_write(&state->games[i], &state->games[i], sizeof(GameState));
          break;
        }
      }
      mmap_write(state, state, sizeof(ServerState));
    }
  }
}

mmap_close(fd, sizeof(ServerState));

stats_init();
printf("\nStatistics updated from server\n");
clear_input_buffer;
```

```c
}

int main() {
  init_random();

  if (!login_menu()) {
    printf("Error logging in\n");
    return 1;
  }

  int running = 1;

  while (running) {
    int choice = main_menu();

    switch (choice) {
      case 1:
        if (create_game_dialog()) {
          OSMmapHandle fd;
          ServerState *state = get_server_state(&fd);
          GameState *game = get_game(state, current_game_id);

          setup_ships(&game->players[player_index].board);
          game->players[player_index].ready = 1;
          mmap_write(game, game, sizeof(GameState));

          printf("\nWaiting for second player...\n");
          printf("(When second player connects, ship placement will begin)\n");

          int waiting = 1;
          while (waiting && game->player_count < 2) {
            os_usleep(500000);
            mmap_read(state, state, sizeof(ServerState));
            game = get_game(state, current_game_id);
            printf(".");
            fflush(stdout);
          }
          printf("\nSecond player connected!\n");

          mmap_close(fd, sizeof(ServerState));
          play_game();
        }
        break;

      case 2:
        if (join_game_dialog()) {
          OSMmapHandle fd;
          ServerState *state = get_server_state(&fd);
          GameState *game = get_game(state, current_game_id);

          setup_ships(&game->players[player_index].board);
          game->players[player_index].ready = 1;
          mmap_write(game, game, sizeof(GameState));

          printf("Waiting for first player...\n");
          int waiting = 1;
          while (waiting && !game->players[0].ready) {
            os_usleep(500000);
            mmap_read(state, state, sizeof(ServerState));
            game = get_game(state, current_game_id);
          }
          printf("Both players ready! Game starts!\n");

          game->status = GAME_RUNNING;
          mmap_write(game, game, sizeof(GameState));
          mmap_close(fd, sizeof(ServerState));
          play_game();
        }
        break;

      case 3:
        stats_init();
        display_player_stats(player_login);
        break;

      case 4:
        running = 0;
        printf("Goodbye!\n");
        break;

      default:
        printf("Invalid choice\n");
    }
  }

  return 0;
}
```

Листинг 14: *Реализация клиента*

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

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

```
1641  17:21:17.018802 brk(NULL)          = 0x55fb185ad000
1641  17:21:17.019025 brk(0x55fb185ce000) = 0x55fb185ce000
1641  17:21:17.019266 write(1,
    "\342\225\220\342\225\220\342\225\220\342\225\220\342\225\220\342\225\220\342\225\220\342\
    118) = 118
1641  17:21:17.019597 write(1, " BATTLESHIP - SERVER \n", 22) = 22
1641  17:21:17.020011 write(1,
    "\342\225\220\342\225\220\342\225\220\342\225\220\342\225\220\342\225\220\342\225\220\342\
    118) = 118
1641  17:21:17.020457 write(1, "\n", 1) = 1
1641  17:21:17.020824 rt_sigaction(SIGINT, {sa_handler=0x55fb08859d72,
    sa_mask=[INT], sa_flags=SA_RESTORER|SA_RESTART,
    sa_restorer=0x7f2d2059a520}, {sa_handler=SIG_DFL, sa_mask=[], sa_flags=0},
    8) = 0
1641  17:21:17.021012 unlink("server_state.mmap") = 0
1641  17:21:17.021931 unlink("server_state.mmap") = -1 ENOENT (No such file or
    directory)
1641  17:21:17.022131 openat(AT_FDCWD, "server_state.mmap",
    O_RDWR|O_CREAT|O_TRUNC, 0666) = 3
1641  17:21:17.022405 lseek(3, 32863, SEEK_SET) = 32863
1641  17:21:17.022568 write(3, "\0", 1) = 1
1641  17:21:17.022763 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0)
    = 0x7f2d2054c000
1641  17:21:17.022963 write(1, "Server initialized\n", 19) = 19
1641  17:21:17.023137 write(1, "Initializing synchronization pri"..., 43) = 43
1641  17:21:17.023504 write(1, "Initialized state_mutex\n", 24) = 24
1641  17:21:17.023925 write(1, "Initialized state_changed condit"..., 45) = 45
1641  17:21:17.024276 write(1, "Synchronization primitives initi"..., 39) = 39
1641  17:21:17.024653 write(1, "\n", 1) = 1
1641  17:21:17.025160 openat(AT_FDCWD, "players_stats.db", O_RDONLY) = 4
1641  17:21:17.025526 newfstatat(4, "", {st_mode=S_IFREG|0644, st_size=76004,
    ...}, AT_EMPTY_PATH) = 0
1641  17:21:17.025709 read(4,
    "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"..., 73728) =
    73728
1641  17:21:17.026705 read(4,
    "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"...,
    4096) = 2276
1641  17:21:17.026985 close(4)          = 0
1641  17:21:17.027337 write(1, "Statistics system initialized\n", 30) = 30
1641  17:21:17.027604 write(1, "\n", 1) = 1
1641  17:21:17.027861 write(1, "Server is running and waiting fo"..., 45) = 45
1641  17:21:17.028132 write(1, "Clients can connect\n", 20) = 20
1641  17:21:17.028500 write(1, "\n", 1) = 1
1641  17:21:17.028804 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:17.538602 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:17.539154 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:17.539480 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:17.539737 close(4)          = 0
```

```
1641  17:21:17.540043 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
1641  17:21:18.041347 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:18.041904 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
       = 0x7f2d20543000
1641  17:21:18.042385 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:18.042727 close(4)          = 0
1641  17:21:18.042911 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
1641  17:21:16.789523 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:16.789834 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
       = 0x7f2d20543000
1641  17:21:16.790177 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:16.790406 close(4)          = 0
1641  17:21:16.790625 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
1641  17:21:17.311394 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:17.312119 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
       = 0x7f2d20543000
1641  17:21:17.312815 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:17.313233 close(4)          = 0
1641  17:21:17.313660 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
1641  17:21:17.825330 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:17.825883 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
       = 0x7f2d20543000
1641  17:21:17.826366 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:17.827114 close(4)          = 0
1641  17:21:17.827486 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
1641  17:21:18.330641 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:18.331094 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
       = 0x7f2d20543000
1641  17:21:18.331493 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:18.331797 close(4)          = 0
1641  17:21:18.332082 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
1641  17:21:18.848632 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:18.849399 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
       = 0x7f2d20543000
1641  17:21:18.849918 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:18.850223 close(4)          = 0
1641  17:21:18.850570 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
1641  17:21:19.387723 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:19.388456 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
       = 0x7f2d20543000
1641  17:21:19.388929 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:19.389362 close(4)          = 0
1641  17:21:19.389734 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
       tv_nsec=500000000}, NULL) = 0
```

```
1641  17:21:19.908950 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:19.909383 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:19.909700 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:19.910014 close(4)          = 0
1641  17:21:19.910291 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:20.429200 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:20.429760 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:20.430098 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:20.430471 close(4)          = 0
1641  17:21:20.430897 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:20.942708 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:20.943495 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:20.943928 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:20.944313 close(4)          = 0
1641  17:21:20.944696 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:21.448271 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:21.448770 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:21.449262 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:21.449562 close(4)          = 0
1641  17:21:21.449974 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:21.951458 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:21.952119 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:21.952726 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:21.953289 close(4)          = 0
1641  17:21:21.953815 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:22.461644 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:22.462256 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:22.462589 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:22.462971 close(4)          = 0
1641  17:21:22.463390 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:22.965734 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:22.966372 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:22.966816 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:22.967297 close(4)          = 0
1641  17:21:22.967761 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:23.474602 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:23.475217 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```
    = 0x7f2d20543000
1641  17:21:23.475806 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:23.476368 close(4)          = 0
1641  17:21:23.476687 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:23.979087 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:23.979808 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:23.980271 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:23.980670 close(4)          = 0
1641  17:21:23.981117 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:24.483085 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:24.483777 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:24.484212 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:24.484539 close(4)          = 0
1641  17:21:24.484918 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:24.987757 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:24.988230 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:24.988642 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:24.989044 close(4)          = 0
1641  17:21:24.989342 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:25.490511 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:25.490918 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:25.491265 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:25.491569 close(4)          = 0
1641  17:21:25.491908 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:25.992623 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:25.993135 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:25.993694 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:25.994216 close(4)          = 0
1641  17:21:25.994554 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:26.502746 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:26.503376 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:26.503682 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:26.503984 close(4)          = 0
1641  17:21:26.504340 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:27.008111 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:27.008571 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:27.009111 munmap(0x7f2d20543000, 32864) = 0
```

```
1641  17:21:27.009492 close(4)          = 0
1641  17:21:27.009922 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:27.511954 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:27.512522 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:27.512963 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:27.513338 close(4)          = 0
1641  17:21:27.513577 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:28.016515 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:28.017165 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:28.017638 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:28.017952 close(4)          = 0
1641  17:21:28.018299 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:28.521974 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:28.522631 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:28.523245 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:28.523863 close(4)          = 0
1641  17:21:28.524428 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:29.026581 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:29.027379 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:29.027757 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:29.028179 close(4)          = 0
1641  17:21:29.028576 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:29.541687 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:29.542433 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:29.543009 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:29.543512 close(4)          = 0
1641  17:21:29.544043 write(1, "\n", 1) = 1
1641  17:21:29.544539 write(1, "[SERVER] Active games: 1/16\n", 28) = 28
1641  17:21:29.544984 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:30.051489 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:30.052038 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:30.052629 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:30.053215 close(4)          = 0
1641  17:21:30.053556 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:30.561665 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:30.562034 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:30.562368 munmap(0x7f2d20543000, 32864) = 0
```

```
1641  17:21:30.562664 close(4)          = 0
1641  17:21:30.563014 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:31.071345 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:31.072075 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:31.072404 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:31.072695 close(4)          = 0
1641  17:21:31.073051 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:31.578975 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:31.579431 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:31.579930 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:31.580255 close(4)          = 0
1641  17:21:31.580641 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:32.084473 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:32.085275 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:32.086024 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:32.086528 close(4)          = 0
1641  17:21:32.086880 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:32.588443 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:32.589094 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:32.589689 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:32.590171 close(4)          = 0
1641  17:21:32.590556 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:33.127229 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:33.127889 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:33.128278 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:33.128618 close(4)          = 0
1641  17:21:33.128895 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:33.634332 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:33.635087 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:33.635531 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:33.635950 close(4)          = 0
1641  17:21:33.636290 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:34.139844 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:34.140743 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:34.141208 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:34.141640 close(4)          = 0
1641  17:21:34.141928 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:34.652087 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:34.652864 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:34.653316 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:34.653748 close(4)          = 0
1641  17:21:34.654099 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:35.157387 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:35.157991 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:35.158424 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:35.158760 close(4)          = 0
1641  17:21:35.159138 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:35.661413 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:35.661836 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:35.662295 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:35.662687 close(4)          = 0
1641  17:21:35.662963 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:36.164439 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:36.165310 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:36.166047 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:36.166509 close(4)          = 0
1641  17:21:36.166862 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:36.669104 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:36.669540 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:36.669916 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:36.670371 close(4)          = 0
1641  17:21:36.670623 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1641  17:21:36.930017 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1641  17:21:36.930202 restart_syscall(<... resuming interrupted clock_nanosleep
    ...>) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1641  17:21:36.930653 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1641  17:21:36.930787 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1641  17:21:36.930921 restart_syscall(<... resuming interrupted restart_syscall
    ...>) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1641  17:21:37.047103 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1641  17:21:37.047433 restart_syscall(<... resuming interrupted restart_syscall
    ...>) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1641  17:21:37.110709 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1641  17:21:37.110912 restart_syscall(<... resuming interrupted restart_syscall
    ...>) = 0
1641  17:21:37.172048 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:37.172688 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```
    = 0x7f2d20543000
1641  17:21:37.173093 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:37.173573 close(4)          = 0
1641  17:21:37.173841 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:37.680013 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:37.680708 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:37.681153 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:37.681580 close(4)          = 0
1641  17:21:37.681871 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:38.182722 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:38.183199 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:38.183461 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:38.183713 close(4)          = 0
1641  17:21:38.183987 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:38.687941 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:38.688467 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:38.688877 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:38.689189 close(4)          = 0
1641  17:21:38.689478 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:39.213018 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:39.213763 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:39.214169 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:39.214563 close(4)          = 0
1641  17:21:39.214915 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:39.735837 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:39.736415 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:39.736891 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:39.737371 close(4)          = 0
1641  17:21:39.737727 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:40.238151 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:40.238515 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:40.238886 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:40.239200 close(4)          = 0
1641  17:21:40.239371 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:40.760794 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:40.761626 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:40.762396 munmap(0x7f2d20543000, 32864) = 0
```

```
1641  17:21:40.762896 close(4)          = 0
1641  17:21:40.763164 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:41.275915 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:41.276576 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:41.277087 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:41.277668 close(4)          = 0
1641  17:21:41.277985 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:41.796146 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:41.796874 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:41.797429 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:41.797929 close(4)          = 0
1641  17:21:41.798461 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:42.298943 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:42.299292 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:42.299871 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:42.300276 close(4)          = 0
1641  17:21:42.300505 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:42.819762 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:42.820483 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:42.820979 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:42.821345 close(4)          = 0
1641  17:21:42.821598 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:43.345733 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:43.346498 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:43.347266 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:43.347723 close(4)          = 0
1641  17:21:43.347977 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:43.871951 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:43.872615 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:43.873163 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:43.873534 close(4)          = 0
1641  17:21:43.873949 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:44.374836 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:44.375243 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:44.375604 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:44.375868 close(4)          = 0
1641  17:21:44.376199 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:44.897970 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:44.898323 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:44.898891 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:44.899405 close(4)          = 0
1641  17:21:44.899781 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:45.401225 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:45.402086 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:45.402567 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:45.402876 close(4)          = 0
1641  17:21:45.403171 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:45.911403 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:45.912036 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:45.912447 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:45.912819 close(4)          = 0
1641  17:21:45.913203 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:46.420887 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:46.421319 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:46.421732 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:46.422077 close(4)          = 0
1641  17:21:46.422462 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:46.929458 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:46.930205 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:46.930511 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:46.930786 close(4)          = 0
1641  17:21:46.931066 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:47.432477 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:47.432961 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:47.433372 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:47.433763 close(4)          = 0
1641  17:21:47.434037 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:47.935757 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:47.936349 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:47.936760 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:47.937152 close(4)          = 0
1641  17:21:47.937660 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:48.441150 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
```

```
1641  17:21:48.441896 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:48.442263 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:48.442604 close(4)          = 0
1641  17:21:48.443009 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:47.081007 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:47.081789 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:47.082297 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:47.082544 close(4)          = 0
1641  17:21:47.082796 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:47.584569 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:47.585165 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:47.585660 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:47.585975 close(4)          = 0
1641  17:21:47.586255 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:48.089657 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:48.090292 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:48.090728 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:48.091164 close(4)          = 0
1641  17:21:48.091516 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:48.594102 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:48.594806 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:48.595189 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:48.595511 close(4)          = 0
1641  17:21:48.595743 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:49.098205 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:49.098648 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:49.099087 write(1, "\n", 1) = 1
1641  17:21:49.099634 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:21:49.100125 write(1, "[SHOT] Player 0 shoots [0,2] -> "..., 63) = 63
1641  17:21:49.100435 write(1, "SUNK [0,2]\n", 11) = 11
1641  17:21:49.100687 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:21:49.101070 write(1, "Result sent (version 2)\n", 24) = 24
1641  17:21:49.101524 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:21:49.101950 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:49.102243 close(4)          = 0
1641  17:21:49.102584 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:49.610013 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:49.610327 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
```

```
1641  17:21:49.610595 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:49.610913 close(4)          = 0
1641  17:21:49.611190 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:50.111862 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:50.112200 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:50.112596 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:50.113012 close(4)          = 0
1641  17:21:50.113387 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:50.615002 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:50.615541 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:50.616063 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:50.616387 close(4)          = 0
1641  17:21:50.616716 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:51.118750 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:51.119346 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:51.119708 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:51.120123 close(4)          = 0
1641  17:21:51.120614 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:51.622966 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:51.623400 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:51.624105 write(1, "\n", 1) = 1
1641  17:21:51.624719 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:21:51.625153 write(1, "[SHOT] Player 0 shoots [1,5] -> "..., 62) = 62
1641  17:21:51.625597 write(1, "HIT [1,5]\n", 10) = 10
1641  17:21:51.626101 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:21:51.626681 write(1, "Result sent (version 4)\n", 24) = 24
1641  17:21:51.627132 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:21:51.627547 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:51.627845 close(4)          = 0
1641  17:21:51.628227 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:52.130647 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:52.131161 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:52.131644 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:52.132172 close(4)          = 0
1641  17:21:52.132505 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:52.635481 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:52.636027 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:52.636336 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:52.636787 close(4)          = 0
```

```
1641  17:21:52.637070 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:53.139184 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:53.139806 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:53.140161 write(1, "\n", 1) = 1
1641  17:21:53.140542 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:21:53.141064 write(1, "[SHOT] Player 0 shoots [1,6] -> "..., 63) = 63
1641  17:21:53.141491 write(1, "SUNK [1,6]\n", 11) = 11
1641  17:21:53.141893 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:21:53.142308 write(1, "Result sent (version 6)\n", 24) = 24
1641  17:21:53.142655 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:21:53.143030 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:53.143340 close(4)          = 0
1641  17:21:53.143636 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:53.663262 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:53.663951 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:53.664345 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:53.664660 close(4)          = 0
1641  17:21:53.664880 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:54.166836 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:54.167418 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:54.167789 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:54.168063 close(4)          = 0
1641  17:21:54.168529 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:54.671250 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:54.671646 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:54.672125 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:54.672385 close(4)          = 0
1641  17:21:54.672661 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:55.174318 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:55.174822 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:55.175314 write(1, "\n", 1) = 1
1641  17:21:55.175853 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:21:55.176246 write(1, "[SHOT] Player 0 shoots [1,9] -> "..., 62) = 62
1641  17:21:55.176582 write(1, "HIT [1,9]\n", 10) = 10
1641  17:21:55.176874 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:21:55.177204 write(1, "Result sent (version 8)\n", 24) = 24
1641  17:21:55.177536 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:21:55.177833 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:55.178327 close(4)          = 0
1641  17:21:55.178775 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
```

```
1641  17:21:55.697931 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:55.698618 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:55.699163 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:55.699452 close(4)          = 0
1641  17:21:55.699782 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:56.201193 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:56.201865 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:56.202160 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:56.202443 close(4)          = 0
1641  17:21:56.202696 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:56.707992 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:56.708725 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:56.709274 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:56.709818 close(4)          = 0
1641  17:21:56.710189 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:57.218512 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:57.219209 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:57.219890 write(1, "\n", 1) = 1
1641  17:21:57.220313 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:21:57.220914 write(1, "[SHOT] Player 0 shoots [2,9] -> "..., 62) = 62
1641  17:21:57.221405 write(1, "HIT [2,9]\n", 10) = 10
1641  17:21:57.221901 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:21:57.222265 write(1, "Result sent (version 10)\n", 25) = 25
1641  17:21:57.222690 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:21:57.223041 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:57.223451 close(4)          = 0
1641  17:21:57.223826 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:57.735287 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:57.735883 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:57.736523 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:57.737064 close(4)          = 0
1641  17:21:57.737420 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:58.245745 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:58.246526 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:58.246987 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:58.247354 close(4)          = 0
1641  17:21:58.247728 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:58.754102 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:58.754671 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```
    = 0x7f2d20543000
1641  17:21:58.755105 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:58.755474 close(4)          = 0
1641  17:21:58.755798 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:59.259912 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:59.260432 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:59.260861 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:59.261326 close(4)          = 0
1641  17:21:59.261988 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:21:59.763441 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:21:59.763944 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:21:59.764457 write(1, "\n", 1) = 1
1641  17:21:59.764967 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:21:59.765439 write(1, "[SHOT] Player 0 shoots [3,1] -> "..., 62) = 62
1641  17:21:59.766022 write(1, "HIT [3,1]\n", 10) = 10
1641  17:21:59.766446 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:21:59.766889 write(1, "Result sent (version 12)\n", 25) = 25
1641  17:21:59.767354 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:21:59.767691 munmap(0x7f2d20543000, 32864) = 0
1641  17:21:59.768174 close(4)          = 0
1641  17:21:59.768461 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:00.270054 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:00.270828 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:00.271231 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:00.271547 close(4)          = 0
1641  17:22:00.271886 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:00.773759 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:00.774508 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:00.774887 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:00.775220 close(4)          = 0
1641  17:22:00.775457 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:01.275995 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:01.276358 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:01.276944 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:01.277374 close(4)          = 0
1641  17:22:01.277957 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:01.779164 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:01.779902 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:01.780649 munmap(0x7f2d20543000, 32864) = 0
```

```
1641  17:22:01.781091 close(4)          = 0
1641  17:22:01.781505 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:02.314158 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:02.314749 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:02.315282 write(1, "\n", 1) = 1
1641  17:22:02.315757 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:02.316278 write(1, "[SHOT] Player 0 shoots [3,3] -> "..., 63) = 63
1641  17:22:02.316704 write(1, "SUNK [3,3]\n", 11) = 11
1641  17:22:02.317086 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:22:02.317504 write(1, "Result sent (version 14)\n", 25) = 25
1641  17:22:02.317833 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:02.318249 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:02.318625 close(4)          = 0
1641  17:22:02.318904 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:02.832356 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:02.833037 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:02.833653 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:02.834032 close(4)          = 0
1641  17:22:02.834346 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:03.336099 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:03.336755 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:03.337335 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:03.337741 close(4)          = 0
1641  17:22:03.338095 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:03.854922 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:03.855371 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:03.855761 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:03.856143 close(4)          = 0
1641  17:22:03.856446 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:04.373398 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:04.374279 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:04.375022 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:04.375388 close(4)          = 0
1641  17:22:04.375691 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:04.891062 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:04.891843 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:04.892561 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:04.893075 close(4)          = 0
1641  17:22:04.893534 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:05.395678 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:05.396396 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:05.396921 write(1, "\n", 1) = 1
1641  17:22:05.397533 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:05.398016 write(1, "[SHOT] Player 0 shoots [3,9] -> "..., 63) = 63
1641  17:22:05.398465 write(1, "SUNK [3,9]\n", 11) = 11
1641  17:22:05.398878 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:22:05.399313 write(1, "Result sent (version 16)\n", 25) = 25
1641  17:22:05.399705 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:05.400087 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:05.400395 close(4)          = 0
1641  17:22:05.400697 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:05.921379 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:05.921921 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:05.922419 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:05.922760 close(4)          = 0
1641  17:22:05.923159 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:06.431929 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:06.432708 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:06.433206 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:06.433575 close(4)          = 0
1641  17:22:06.433928 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:06.935781 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:06.936484 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:06.936956 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:06.937443 close(4)          = 0
1641  17:22:06.937848 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:07.439175 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:07.439621 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:07.440095 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:07.440505 close(4)          = 0
1641  17:22:07.440873 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:07.964616 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:07.965266 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:07.965726 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:07.966053 close(4)          = 0
1641  17:22:07.966371 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:08.506007 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
```

```
1641  17:22:08.506818 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:08.507496 write(1, "\n", 1) = 1
1641  17:22:08.507900 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:08.508300 write(1, "[SHOT] Player 0 shoots [4,1] -> "..., 63) = 63
1641  17:22:08.508757 write(1, "SUNK [4,1]\n", 11) = 11
1641  17:22:08.509295 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:22:08.509640 write(1, "Result sent (version 18)\n", 25) = 25
1641  17:22:08.510063 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:08.510375 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:08.510835 close(4)          = 0
1641  17:22:08.511240 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:09.012818 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:09.013579 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:09.014098 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:09.014433 close(4)          = 0
1641  17:22:09.014700 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:09.516110 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:09.516777 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:09.517181 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:09.517525 close(4)          = 0
1641  17:22:09.517914 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:10.019115 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:10.019757 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:10.020326 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:10.020799 close(4)          = 0
1641  17:22:10.021241 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:10.528841 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:10.529551 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:10.530137 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:10.530573 close(4)          = 0
1641  17:22:10.530867 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:11.032592 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:11.033298 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:11.033810 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:11.034283 close(4)          = 0
1641  17:22:11.034728 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:11.535369 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:11.535907 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
```

```
1641  17:22:11.536411 write(1, "\n", 1) = 1
1641  17:22:11.536688 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:11.537085 write(1, "[SHOT] Player 0 shoots [5,6] -> "..., 62) = 62
1641  17:22:11.537450 write(1, "HIT [5,6]\n", 10) = 10
1641  17:22:11.537727 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:22:11.538040 write(1, "Result sent (version 20)\n", 25) = 25
1641  17:22:11.538370 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:11.538638 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:11.538955 close(4)          = 0
1641  17:22:11.539189 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:12.042297 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:12.042972 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:12.043448 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:12.043851 close(4)          = 0
1641  17:22:12.044219 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:12.554604 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:12.555232 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:12.555821 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:12.556297 close(4)          = 0
1641  17:22:12.556689 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:13.057947 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:13.058819 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:13.059201 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:13.059649 close(4)          = 0
1641  17:22:13.060109 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:13.562870 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:13.563354 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:13.563878 write(1, "\n", 1) = 1
1641  17:22:13.564439 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:13.564964 write(1, "[SHOT] Player 0 shoots [5,7] -> "..., 63) = 63
1641  17:22:13.565407 write(1, "SUNK [5,7]\n", 11) = 11
1641  17:22:13.565749 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:22:13.566075 write(1, "Result sent (version 22)\n", 25) = 25
1641  17:22:13.566347 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:13.566645 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:13.566890 close(4)          = 0
1641  17:22:13.567132 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:14.072765 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:14.073252 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:14.073644 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:14.073999 close(4)          = 0
```

```
1641  17:22:14.074361 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:14.575431 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:14.575958 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:14.576466 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:14.576902 close(4)          = 0
1641  17:22:14.577233 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:15.101205 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:15.101813 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:15.102321 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:15.102717 close(4)          = 0
1641  17:22:15.103053 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:15.604815 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:15.605337 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:15.605817 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:15.606184 close(4)          = 0
1641  17:22:15.606601 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:16.109853 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:16.110523 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:16.111225 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:16.111657 close(4)          = 0
1641  17:22:16.112025 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:16.613242 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:16.613824 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:16.614441 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:16.614721 close(4)          = 0
1641  17:22:16.615101 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:17.140766 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:17.141313 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:17.141629 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:17.142014 close(4)          = 0
1641  17:22:17.142371 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:17.643638 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:17.644196 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:17.644556 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:17.644974 close(4)          = 0
1641  17:22:17.645344 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
```

```
1641  17:22:18.148945 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:18.149586 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:18.150031 write(1, "\n", 1) = 1
1641  17:22:18.150343 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:18.150818 write(1, "[SHOT] Player 0 shoots [6,1] -> "..., 62) = 62
1641  17:22:18.151306 write(1, "HIT [6,1]\n", 10) = 10
1641  17:22:18.151832 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:22:18.152230 write(1, "Result sent (version 24)\n", 25) = 25
1641  17:22:18.152624 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:18.152991 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:18.153283 close(4)          = 0
1641  17:22:18.153516 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:18.654606 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:18.654995 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:18.655476 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:18.655888 close(4)          = 0
1641  17:22:18.656369 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:17.343723 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:17.344293 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:17.344631 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:17.344980 close(4)          = 0
1641  17:22:17.345488 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:17.859690 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:17.860464 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:17.861019 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:17.861387 close(4)          = 0
1641  17:22:17.861858 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:18.362544 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:18.363134 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:18.363685 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:18.364017 close(4)          = 0
1641  17:22:18.364433 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:18.875469 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:18.875947 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:18.876255 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:18.876565 close(4)          = 0
1641  17:22:18.876857 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:19.378024 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:19.378782 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```
    = 0x7f2d20543000
1641  17:22:19.379365 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:19.379769 close(4)          = 0
1641  17:22:19.380133 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:19.883744 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:19.884406 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:19.884914 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:19.885351 close(4)          = 0
1641  17:22:19.885825 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:20.387301 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:20.388039 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:20.388639 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:20.389044 close(4)          = 0
1641  17:22:20.389427 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:20.892165 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:20.892738 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:20.893278 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:20.893676 close(4)          = 0
1641  17:22:20.894001 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:21.395579 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:21.396380 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:21.396864 write(1, "\n", 1) = 1
1641  17:22:21.397314 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:21.397697 write(1, "[SHOT] Player 0 shoots [6,2] -> "..., 62) = 62
1641  17:22:21.398092 write(1, "HIT [6,2]\n", 10) = 10
1641  17:22:21.398479 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:22:21.399011 write(1, "Result sent (version 26)\n", 25) = 25
1641  17:22:21.399484 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:21.399838 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:21.400115 close(4)          = 0
1641  17:22:21.400399 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:21.904610 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:21.905880 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:21.906282 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:21.906574 close(4)          = 0
1641  17:22:21.906903 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:22.408063 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:22.408590 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:22.409019 munmap(0x7f2d20543000, 32864) = 0
```

```
1641  17:22:22.409304 close(4)          = 0
1641  17:22:22.409572 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=500000000}, NULL) = 0
1641  17:22:22.917533 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:22.918192 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f2d20543000
1641  17:22:22.918629 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:22.919044 close(4)          = 0
1641  17:22:22.919439 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=500000000}, NULL) = 0
1641  17:22:23.421324 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:23.421969 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f2d20543000
1641  17:22:23.422480 write(1, "\n", 1) = 1
1641  17:22:23.422885 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:23.423172 write(1, "[SHOT] Player 0 shoots [6,3] -> "..., 63) = 63
1641  17:22:23.423560 write(1, "SUNK [6,3]\n", 11) = 11
1641  17:22:23.423963 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:22:23.424531 write(1, "Result sent (version 28)\n", 25) = 25
1641  17:22:23.424967 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:23.425382 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:23.425795 close(4)          = 0
1641  17:22:23.426202 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=500000000}, NULL) = 0
1641  17:22:23.928629 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:23.929318 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f2d20543000
1641  17:22:23.929903 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:23.930367 close(4)          = 0
1641  17:22:23.930769 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=500000000}, NULL) = 0
1641  17:22:24.432742 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:24.433232 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f2d20543000
1641  17:22:24.433831 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:24.434232 close(4)          = 0
1641  17:22:24.434696 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=500000000}, NULL) = 0
1641  17:22:24.940698 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:24.941239 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f2d20543000
1641  17:22:24.941857 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:24.942201 close(4)          = 0
1641  17:22:24.942495 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=500000000}, NULL) = 0
1641  17:22:25.443971 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:25.444890 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f2d20543000
1641  17:22:25.445289 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:25.445600 close(4)          = 0
1641  17:22:25.446014 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:25.955721 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:25.956482 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:25.956941 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:25.957231 close(4)          = 0
1641  17:22:25.957542 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:26.466486 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:26.466918 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:26.467466 write(1, "\n", 1) = 1
1641  17:22:26.467940 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:26.468378 write(1, "[SHOT] Player 0 shoots [7,8] -> "..., 63) = 63
1641  17:22:26.468816 write(1, "SUNK [7,8]\n", 11) = 11
1641  17:22:26.469186 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:22:26.469610 write(1, "Result sent (version 30)\n", 25) = 25
1641  17:22:26.470033 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:26.470347 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:26.470651 close(4)          = 0
1641  17:22:26.470964 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:26.979726 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:26.980367 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:26.980995 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:26.981279 close(4)          = 0
1641  17:22:26.981555 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:27.484309 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:27.484782 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:27.485210 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:27.485618 close(4)          = 0
1641  17:22:27.485963 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:28.006492 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:28.007209 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:28.007862 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:28.008305 close(4)          = 0
1641  17:22:28.008727 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:28.510573 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:28.511512 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:28.511975 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:28.512390 close(4)          = 0
1641  17:22:28.512671 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:29.016456 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
```

```
1641  17:22:29.017130 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:29.017597 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:29.018092 close(4)          = 0
1641  17:22:29.018391 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:29.523191 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:29.524092 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:29.524512 write(1, "\n", 1) = 1
1641  17:22:29.524875 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:29.525384 write(1, "[SHOT] Player 0 shoots [8,0] -> "..., 63) = 63
1641  17:22:29.525817 write(1, "SUNK [8,0]\n", 11) = 11
1641  17:22:29.526192 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:22:29.526585 write(1, "Result sent (version 32)\n", 25) = 25
1641  17:22:29.526940 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:29.527216 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:29.527529 close(4)          = 0
1641  17:22:29.527862 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:30.043227 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:30.043969 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:30.044454 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:30.044767 close(4)          = 0
1641  17:22:30.045037 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:30.549731 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:30.550583 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:30.551185 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:30.551462 close(4)          = 0
1641  17:22:30.551842 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:31.058657 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:31.059161 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:31.059510 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:31.059909 close(4)          = 0
1641  17:22:31.060298 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:31.591525 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:31.592164 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:31.592634 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:31.593133 close(4)          = 0
1641  17:22:31.593722 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:32.106783 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:32.107481 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
```

```
1641  17:22:32.107916 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:32.108202 close(4)          = 0
1641  17:22:32.108512 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:32.610190 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:32.611003 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:32.611620 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:32.612034 close(4)          = 0
1641  17:22:32.612410 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:33.123710 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:33.124403 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:33.124845 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:33.125155 close(4)          = 0
1641  17:22:33.125451 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:33.637008 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:33.637698 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:33.638280 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:33.638641 close(4)          = 0
1641  17:22:33.638946 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:34.147241 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:34.147940 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:34.148326 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:34.148628 close(4)          = 0
1641  17:22:34.148923 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:34.650132 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:34.650474 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:34.650930 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:34.651289 close(4)          = 0
1641  17:22:34.651635 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:35.164552 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:35.165240 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:35.165691 write(1, "\n", 1) = 1
1641  17:22:35.166105 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:35.166475 write(1, "[SHOT] Player 0 shoots [9,5] -> "..., 62) = 62
1641  17:22:35.166846 write(1, "HIT [9,5]\n", 10) = 10
1641  17:22:35.167296 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:22:35.167863 write(1, "Result sent (version 34)\n", 25) = 25
1641  17:22:35.168208 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:35.168487 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:35.168799 close(4)          = 0
```

```
1641  17:22:35.169143 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:35.689625 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:35.690428 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:35.691132 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:35.691630 close(4)           = 0
1641  17:22:35.691955 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:36.193098 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:36.193898 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:36.194433 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:36.194719 close(4)           = 0
1641  17:22:36.194991 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:36.696333 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:36.696988 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:36.697318 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:36.697835 close(4)           = 0
1641  17:22:36.698259 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:37.214641 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:37.215462 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:37.215913 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:37.216239 close(4)           = 0
1641  17:22:37.216697 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:37.748095 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:37.748769 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:37.749289 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:37.749727 close(4)           = 0
1641  17:22:37.750160 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:38.250836 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:38.251151 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:38.251436 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:38.251692 close(4)           = 0
1641  17:22:38.251948 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:38.752598 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:38.752900 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:38.753110 write(1, "\n", 1) = 1
1641  17:22:38.753436 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:38.753744 write(1, "[SHOT] Player 0 shoots [9,4] -> "..., 62) = 62
1641  17:22:38.754126 write(1, "HIT [9,4]\n", 10) = 10
```

```
1641  17:22:38.754410 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:22:38.754684 write(1, "Result sent (version 36)\n", 25) = 25
1641  17:22:38.754877 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:38.755099 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:38.755292 close(4)          = 0
1641  17:22:38.755485 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:39.273922 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:39.274615 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:39.275097 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:39.275416 close(4)          = 0
1641  17:22:39.275740 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:39.798908 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:39.799582 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:39.800152 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:39.800460 close(4)          = 0
1641  17:22:39.800810 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:40.303603 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:40.303959 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:40.304459 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:40.305016 close(4)          = 0
1641  17:22:40.305411 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:40.806372 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:40.806665 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:40.807109 write(1, "\n", 1) = 1
1641  17:22:40.807532 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:40.808037 write(1, "[SHOT] Player 0 shoots [9,6] -> "..., 62) = 62
1641  17:22:40.808358 write(1, "HIT [9,6]\n", 10) = 10
1641  17:22:40.808771 futex(0x7f2d2054c7f8, FUTEX_WAKE, 2147483647) = 2
1641  17:22:40.809147 write(1, "Result sent (version 38)\n", 25) = 25
1641  17:22:40.809544 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:40.809845 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:40.810023 close(4)          = 0
1641  17:22:40.810214 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:41.320877 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:41.321114 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:41.321551 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:41.321815 close(4)          = 0
1641  17:22:41.321985 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:41.827772 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:41.828261 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```
    = 0x7f2d20543000
1641  17:22:41.828767 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:41.829176 close(4)          = 0
1641  17:22:41.829544 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:42.333230 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:42.333835 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:42.334276 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:42.334628 close(4)          = 0
1641  17:22:42.334934 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:42.835613 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:42.835962 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:42.836392 write(1, "\n", 1) = 1
1641  17:22:42.836899 write(1, "[GAME: lol] Processing shot from"..., 42) = 42
1641  17:22:42.837409 write(1, "[SHOT] Player 0 shoots [9,7] -> "..., 63) = 63
1641  17:22:42.837942 write(1, "[VICTORY] Player 0 won!\n", 24) = 24
1641  17:22:42.838352 write(1, "*** VICTORY! Player 0 won!\n", 27) = 27
1641  17:22:42.838842 futex(0x7f2d2054c7fc, FUTEX_WAKE, 2147483647) = 2
1641  17:22:42.839485 write(1, "Result sent (version 40)\n", 25) = 25
1641  17:22:42.839991 write(1, "\n", 1) = 1
1641  17:22:42.840398 write(1, "[GAME lol] Finished!\n", 21) = 21
1641  17:22:42.840720 write(1, " Winner: kolko\n", 15) = 15
1641  17:22:42.841011 write(1, " Loser: dimas\n", 14) = 14
1641  17:22:42.841334 openat(AT_FDCWD, "players_stats.db",
    O_WRONLY|O_CREAT|O_TRUNC, 0666) = 5
1641  17:22:42.841734 newfstatat(5, "", {st_mode=S_IFREG|0644, st_size=0, ...},
    AT_EMPTY_PATH) = 0
1641  17:22:42.842123 write(5,
    "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"..., 73728) =
    73728
1641  17:22:42.842830 write(5,
    "\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"...,
    2276) = 2276
1641  17:22:42.843089 close(5)          = 0
1641  17:22:42.843689 openat(AT_FDCWD, "players_stats.db",
    O_WRONLY|O_CREAT|O_TRUNC, 0666) = 5
1641  17:22:42.845016 newfstatat(5, "", {st_mode=S_IFREG|0644, st_size=0, ...},
    AT_EMPTY_PATH) = 0
1641  17:22:42.845363 write(5,
    "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"..., 73728) =
    73728
1641  17:22:42.845970 write(5,
    "\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"...,
    2276) = 2276
1641  17:22:42.846293 close(5)          = 0
1641  17:22:42.846932 futex(0x7f2d2054c7a8, FUTEX_WAKE, 1) = 1
1641  17:22:42.847222 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:42.847520 close(4)          = 0
```

```
1641  17:22:42.847699 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:43.349044 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:43.349809 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:43.350518 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:43.350945 close(4)          = 0
1641  17:22:43.351267 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:43.856727 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:43.857146 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:43.857667 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:43.858164 close(4)          = 0
1641  17:22:43.858553 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:44.359417 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:44.359794 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:44.360098 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:44.360416 close(4)          = 0
1641  17:22:44.360851 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:44.867261 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:44.867749 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:44.868309 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:44.868776 close(4)          = 0
1641  17:22:44.869218 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:45.374290 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:45.375023 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:45.375531 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:45.375990 close(4)          = 0
1641  17:22:45.376386 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:45.885522 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:45.886405 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:45.887186 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:45.887720 close(4)          = 0
1641  17:22:45.888170 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:46.392853 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:46.393605 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:46.394159 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:46.394556 close(4)          = 0
1641  17:22:46.394943 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
```

```
1641  17:22:46.904342 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:46.905038 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:46.905495 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:46.905941 close(4)          = 0
1641  17:22:46.906276 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:47.409673 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:47.410082 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:47.410475 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:47.410757 close(4)          = 0
1641  17:22:47.411010 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:47.920557 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:47.921343 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:47.922061 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:47.922720 close(4)          = 0
1641  17:22:47.923048 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:48.424989 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:48.425303 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:48.425612 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:48.425950 close(4)          = 0
1641  17:22:48.426301 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:48.930935 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:48.931614 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:48.932272 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:48.932690 close(4)          = 0
1641  17:22:48.933077 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:49.434374 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:49.434622 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:49.435018 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:49.435619 close(4)          = 0
1641  17:22:49.436046 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:48.174989 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:48.175649 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:48.176069 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:48.176439 close(4)          = 0
1641  17:22:48.176757 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:48.679050 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:48.679308 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
```

```
    = 0x7f2d20543000
1641  17:22:48.679563 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:48.679855 close(4)          = 0
1641  17:22:48.680079 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:49.184478 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:49.185221 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:49.185594 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:49.185981 close(4)          = 0
1641  17:22:49.186355 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1641  17:22:49.692015 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:49.692315 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:49.692713 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:49.693009 close(4)          = 0
1641  17:22:49.693384 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1641  17:22:50.194038 --- SIGINT {si_signo=SIGINT, si_code=SI_KERNEL} ---
1641  17:22:50.194260 rt_sigreturn({mask=[]}) = -1 EINTR (Interrupted system
    call)
1641  17:22:50.194727 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 4
1641  17:22:50.195336 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0)
    = 0x7f2d20543000
1641  17:22:50.195823 munmap(0x7f2d20543000, 32864) = 0
1641  17:22:50.196185 close(4)          = 0
1641  17:22:50.196664 write(1, "\n", 1) = 1
1641  17:22:50.197153 write(1, "[SERVER] Shutting down...\n", 26) = 26
1641  17:22:50.197672 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=5, tv_nsec=0},
    NULL) = ? ERESTART_RESTARTBLOCK (Interrupted by signal)
1641  17:22:53.077456 --- SIGINT {si_signo=SIGINT, si_code=SI_KERNEL} ---
1641  17:22:53.077646 rt_sigreturn({mask=[]}) = -1 EINTR (Interrupted system
    call)
1641  17:22:53.078158 write(1, "Synchronization primitives clean"..., 38) = 38
1641  17:22:53.078521 munmap(0x2, 32864) = -1 EINVAL (Invalid argument)
1641  17:22:53.078848 close(395049983)  = -1 EBADF (Bad file descriptor)
1641  17:22:53.079210 write(1, "Server stopped\n", 15) = 15
1641  17:22:53.079740 exit_group(0)     = ?
1641  17:22:53.080221 +++ exited with 0 +++
```

Листинг 15: *Strace логи сервера*

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

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

```
1674  17:21:16.795335 getpid()          = 1674
1674  17:21:16.795723 newfstatat(1, "", {st_mode=S_IFCHR|0620,
    st_rdev=makedev(0x88, 0x5), ...}, AT_EMPTY_PATH) = 0
1674  17:21:16.796087 getrandom("\xcf\x15\xfe\x02\x2e\x8f\x5a\x2b", 8,
    GRND_NONBLOCK) = 8
1674  17:21:16.796334 brk(NULL)         = 0x558c9ac97000
1674  17:21:16.796535 brk(0x558c9acb8000) = 0x558c9acb8000
1674  17:21:16.796757 write(1, "\n", 1) = 1
1674  17:21:16.797081 write(1, "================================"..., 43) = 43
1674  17:21:16.797355 write(1, "=  BATTLESHIP - CLIENT          "..., 42) = 42
1674  17:21:16.797683 write(1, "================================"..., 43) = 43
1674  17:21:16.797923 write(1, "\n", 1) = 1
1674  17:21:16.798213 newfstatat(0, "", {st_mode=S_IFCHR|0620,
    st_rdev=makedev(0x88, 0x5), ...}, AT_EMPTY_PATH) = 0
1674  17:21:16.798477 write(1, "Enter your login (up to 63 chara"..., 40) = 40
1674  17:21:16.798693 read(0, "kolko\n", 1024) = 6
1674  17:21:26.211808 write(1, "Welcome, kolko!\n", 16) = 16
1674  17:21:26.212233 openat(AT_FDCWD, "players_stats.db", O_RDONLY) = 3
1674  17:21:26.212565 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=76004,
    ...}, AT_EMPTY_PATH) = 0
1674  17:21:26.212864 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"..., 73728) =
    73728
1674  17:21:26.213255 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
1674  17:21:26.213469 close(3)          = 0
1674  17:21:26.213770 write(1, "\n", 1) = 1
1674  17:21:26.214127 write(1, "========== MAIN MENU ==========\n", 32) = 32
1674  17:21:26.214640 write(1, "1. Create new game\n", 19) = 19
1674  17:21:26.215038 write(1, "2. Join a game\n", 15) = 15
1674  17:21:26.215363 write(1, "3. View my statistics\n", 22) = 22
1674  17:21:26.215681 write(1, "4. Exit\n", 8) = 8
1674  17:21:26.215976 write(1, "Choose action (1-4): ", 21) = 21
1674  17:21:26.216243 read(0, "1\n", 1024) = 2
1674  17:21:28.109095 write(1, "\n", 1) = 1
1674  17:21:28.109619 write(1, "Enter name of new game: ", 24) = 24
1674  17:21:28.110114 read(0, "lol\n", 1024) = 4
1674  17:21:29.228788 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 3
1674  17:21:29.229251 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0)
    = 0x7f7fa8a9c000
1674  17:21:29.229633 write(1, "Game 'lol' created (ID: 0)\n", 27) = 27
1674  17:21:29.230171 munmap(0x7f7fa8a9c000, 32864) = 0
1674  17:21:29.230496 close(3)          = 0
1674  17:21:29.230801 write(1, "Game created! ID: 0\n", 20) = 20
1674  17:21:29.231191 write(1, "Waiting for second player...\n", 29) = 29
1674  17:21:29.231583 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 3
1674  17:21:29.232054 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0)
    = 0x7f7fa8a9c000
1674  17:21:29.232499 write(1, "\n", 1) = 1
1674  17:21:29.233060 write(1, "========== SHIP PLACEMENT ======"..., 37) = 37
```

```
1674  17:21:29.233419 write(1, "Generating random ship placement"..., 36) = 36
1674  17:21:29.233828 write(1, "- 1 four-deck ship (1x4)\n", 25) = 25
1674  17:21:29.234298 write(1, "- 2 three-deck ships (2x3)\n", 27) = 27
1674  17:21:29.234813 write(1, "- 3 two-deck ships (3x2)\n", 25) = 25
1674  17:21:29.235273 write(1, "- 4 one-deck ships (4x1)\n", 25) = 25
1674  17:21:29.235779 write(1, "\n", 1) = 1
1674  17:21:29.236211 write(1, "\n", 1) = 1
1674  17:21:29.236668 write(1, "Ships placed randomly!\n", 23) = 23
1674  17:21:29.237129 write(1, "   0 1 2 3 4 5 6 7 8 9\n", 23) = 23
1674  17:21:29.237491 write(1, " 0 . S . . . . . S S . \n", 24) = 24
1674  17:21:29.237870 write(1, " 1 . . . . . . . . . . \n", 24) = 24
1674  17:21:29.238400 write(1, " 2 S . . . . . S S . . \n", 24) = 24
1674  17:21:29.238784 write(1, " 3 . . . . . . . . . . \n", 24) = 24
1674  17:21:29.239105 write(1, " 4 S S S S . . . . . . \n", 24) = 24
1674  17:21:29.239479 write(1, " 5 . . . . . . S S . . \n", 24) = 24
1674  17:21:29.239808 write(1, " 6 . S . S . . . . . . \n", 24) = 24
1674  17:21:29.240186 write(1, " 7 . . . . . . . . . . \n", 24) = 24
1674  17:21:29.240619 write(1, " 8 S S S . . . S S S . \n", 24) = 24
1674  17:21:29.240963 write(1, " 9 . . . . . . . . . . \n", 24) = 24
1674  17:21:29.241336 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:29.243836 write(1, "\n", 1) = 1
1674  17:21:29.244256 write(1, "Waiting for second player...\n", 29) = 29
1674  17:21:29.244710 write(1, "(When second player connects, sh"..., 57) = 57
1674  17:21:29.245312 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:29.770176 write(1, ".", 1)  = 1
1674  17:21:29.770843 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:30.271833 write(1, ".", 1)  = 1
1674  17:21:30.272341 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:30.784226 write(1, ".", 1)  = 1
1674  17:21:30.784667 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:31.286341 write(1, ".", 1)  = 1
1674  17:21:31.287052 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:31.797359 write(1, ".", 1)  = 1
1674  17:21:31.797861 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:32.305280 write(1, ".", 1)  = 1
1674  17:21:32.305783 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:32.814120 write(1, ".", 1)  = 1
1674  17:21:32.814730 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:33.320322 write(1, ".", 1)  = 1
1674  17:21:33.321190 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:33.825454 write(1, ".", 1)  = 1
1674  17:21:33.826403 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:34.330201 write(1, ".", 1)  = 1
1674  17:21:34.331065 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:34.835592 write(1, ".", 1)  = 1
1674  17:21:34.836102 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:35.340098 write(1, ".", 1)  = 1
1674  17:21:35.340748 clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0,
    tv_nsec=500000000}, NULL) = 0
1674  17:21:35.844981 write(1, ".", 1)  = 1
1674  17:21:35.845484 write(1, "\n", 1) = 1
1674  17:21:35.846172 write(1, "Second player connected!\n", 25) = 25
1674  17:21:35.846594 munmap(0x7f7fa8a9c000, 32864) = 0
1674  17:21:35.847152 close(3)          = 0
1674  17:21:35.847633 write(1, "\n", 1) = 1
1674  17:21:35.848141 write(1, "========== GAME START =========="..., 33) = 33
1674  17:21:35.848598 openat(AT_FDCWD, "server_state.mmap", O_RDWR) = 3
1674  17:21:35.849120 mmap(NULL, 32864, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0)
    = 0x7f7fa8a9c000
1674  17:21:35.849511 write(1, "\n", 1) = 1
1674  17:21:35.849968 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:35.850318 write(1, "\n", 1) = 1
1674  17:21:35.850582 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:35.851049 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:35.851383 write(1, " 0 . S . . . . . S S .   0 . . ."..., 48) = 48
1674  17:21:35.851706 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:35.852167 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:35.852720 write(1, " 3 . . . . . . . . . .   3 . . ."..., 48) = 48
1674  17:21:35.853205 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:35.853659 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:35.854122 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:35.854502 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:35.854900 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:35.855300 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:35.855616 write(1, "\n", 1) = 1
1674  17:21:35.855879 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:21:35.856215 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:35.856456 read(0, 0x558c9ac976b0, 1024) = ? ERESTARTSYS (To be
    restarted if SA_RESTART is set)
1674  17:21:36.930541 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1674  17:21:36.930699 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1674  17:21:36.930933 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1674  17:21:36.931157 read(0, 0x558c9ac976b0, 1024) = ? ERESTARTSYS (To be
    restarted if SA_RESTART is set)
1674  17:21:37.047504 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1674  17:21:37.047681 read(0, 0x558c9ac976b0, 1024) = ? ERESTARTSYS (To be
    restarted if SA_RESTART is set)
1674  17:21:37.111059 --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
1674  17:21:37.111248 read(0, "0 22\n", 1024) = 5
1674  17:21:47.467630 write(1, "Error: coordinates out of bounds"..., 39) = 39
```

```
1674  17:21:47.468126 write(1, "\n", 1) = 1
1674  17:21:47.468590 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:47.468918 write(1, "\n", 1) = 1
1674  17:21:47.469208 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:47.469533 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:47.469856 write(1, " 0 . S . . . . . S S .   0 . . ."..., 48) = 48
1674  17:21:47.470141 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:47.470413 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:47.470688 write(1, " 3 . . . . . . . . . .   3 . . ."..., 48) = 48
1674  17:21:47.470929 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:47.471189 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:47.471419 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:47.471642 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:47.471864 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:47.472086 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:47.472367 write(1, "\n", 1) = 1
1674  17:21:47.472607 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:21:47.472822 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:47.473008 read(0, "0 2\n", 1024) = 4
1674  17:21:48.647430 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:21:48.647839 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:48.649909 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:48.651837 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:21:49.101021 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:21:49.101955 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:21:49.102442 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:21:49.102752 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:49.105133 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:49.107340 write(1, "\n", 1) = 1
1674  17:21:49.107669 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:49.107994 write(1, "\n", 1) = 1
1674  17:21:49.108338 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:49.108662 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:49.109117 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:21:49.109529 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:49.109926 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:49.110362 write(1, " 3 . . . . . . . . . .   3 . . ."..., 48) = 48
1674  17:21:49.110762 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:49.111298 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:49.111774 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:49.112059 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:49.112400 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:49.112767 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:49.113101 write(1, "\n", 1) = 1
1674  17:21:49.113392 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:21:49.113774 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:49.114130 read(0, "1 5\n", 1024) = 4
1674  17:21:51.490802 write(1, "...Sending shot to server...\n", 29) = 29
```

```
1674  17:21:51.491262 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:51.494255 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:51.497407 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:21:51.626626 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:21:51.627706 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:21:51.628047 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:21:51.628395 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:51.630681 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:51.631743 write(1, "\n", 1) = 1
1674  17:21:51.632163 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:51.632578 write(1, "\n", 1) = 1
1674  17:21:51.633001 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:51.633395 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:51.633718 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:21:51.634102 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:51.634528 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:51.634867 write(1, " 3 . . . . . . . . . .   3 . . ."..., 48) = 48
1674  17:21:51.635178 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:51.635494 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:51.635863 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:51.636215 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:51.636550 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:51.636900 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:51.637296 write(1, "\n", 1) = 1
1674  17:21:51.637797 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:21:51.638204 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:51.638515 read(0, "1 6\n", 1024) = 4
1674  17:21:52.640637 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:21:52.640962 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:52.643548 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:52.645547 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:21:53.142269 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:21:53.143130 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:21:53.143465 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:21:53.143890 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:53.146923 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:53.148219 write(1, "\n", 1) = 1
1674  17:21:53.148686 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:53.149070 write(1, "\n", 1) = 1
1674  17:21:53.149450 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:53.149885 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:53.150372 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:21:53.150770 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:53.151129 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:53.151528 write(1, " 3 . . . . . . . . . .   3 . . ."..., 48) = 48
```

```
1674  17:21:53.151901 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:53.152359 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:53.152915 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:53.153462 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:53.153808 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:53.154200 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:53.154621 write(1, "\n", 1) = 1
1674  17:21:53.155082 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:21:53.155516 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:53.155959 read(0, "1 9\n", 1024) = 4
1674  17:21:54.684485 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:21:54.684969 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:54.687239 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:54.689502 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:21:55.177210 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:21:55.177988 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:21:55.178631 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:21:55.179097 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:55.182045 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:55.184015 write(1, "\n", 1) = 1
1674  17:21:55.184611 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:55.185033 write(1, "\n", 1) = 1
1674  17:21:55.185390 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:55.185746 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:55.186062 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:21:55.186435 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:55.186885 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:55.187328 write(1, " 3 . . . . . . . . . .   3 . . ."..., 48) = 48
1674  17:21:55.187823 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:55.188300 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:55.188714 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:55.189090 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:55.189437 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:55.189741 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:55.190060 write(1, "\n", 1) = 1
1674  17:21:55.190388 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:21:55.190730 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:55.191085 read(0, "2 9\n", 1024) = 4
1674  17:21:57.104380 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:21:57.104785 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:57.108098 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:57.111502 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:21:57.222503 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = -1 EAGAIN
    (Resource temporarily unavailable)
1674  17:21:57.222903 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
```

```
1674  17:21:57.223250 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:21:57.223567 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:57.226861 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:57.228643 write(1, "\n", 1) = 1
1674  17:21:57.229254 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:57.229689 write(1, "\n", 1) = 1
1674  17:21:57.230032 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:57.230485 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:57.230893 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:21:57.231345 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:57.231715 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:57.232071 write(1, " 3 . . . . . . . . . .   3 . . ."..., 48) = 48
1674  17:21:57.232472 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:57.232873 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:57.233226 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:57.233515 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:57.233796 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:57.234066 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:57.234375 write(1, "\n", 1) = 1
1674  17:21:57.234783 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:21:57.235213 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:57.235514 read(0, "3 1\n", 1024) = 4
1674  17:21:59.676857 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:21:59.677221 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:59.680248 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:59.682245 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:21:59.766936 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:21:59.767895 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:21:59.768229 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:21:59.768632 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:21:59.772418 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:21:59.773397 write(1, "\n", 1) = 1
1674  17:21:59.773812 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:21:59.774270 write(1, "\n", 1) = 1
1674  17:21:59.774779 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:21:59.775387 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:21:59.775847 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:21:59.776232 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:21:59.776557 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:21:59.776940 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:21:59.777250 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:21:59.777574 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:21:59.777931 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:21:59.778224 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:21:59.778556 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:21:59.778961 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:21:59.779250 write(1, "\n", 1) = 1
1674  17:21:59.779634 write(1, "=== YOUR TURN ===\n", 18) = 18
```

```
1674  17:21:59.780095 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:21:59.780516 read(0, "3 3\n", 1024) = 4
1674  17:22:01.969502 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:01.970007 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:01.973048 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:01.974957 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:02.317633 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:02.318393 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:02.318814 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:22:02.319213 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:02.323251 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:02.324406 write(1, "\n", 1) = 1
1674  17:22:02.324971 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:02.325317 write(1, "\n", 1) = 1
1674  17:22:02.325684 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:02.326067 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:02.326565 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:02.327004 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:02.327382 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:02.327789 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:02.328211 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:22:02.328612 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:02.328937 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:22:02.329259 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:02.329718 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:02.330117 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:02.330491 write(1, "\n", 1) = 1
1674  17:22:02.330928 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:02.331404 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:02.331852 read(0, "3 9\n", 1024) = 4
1674  17:22:05.000937 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:05.001299 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:05.003325 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:05.005198 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:05.399313 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:05.400139 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:05.400451 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:22:05.400761 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:05.403377 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:05.406068 write(1, "\n", 1) = 1
1674  17:22:05.406644 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:05.407191 write(1, "\n", 1) = 1
1674  17:22:05.407552 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:05.407935 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:05.408503 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
```

```
1674  17:22:05.409005 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:05.409380 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:05.409687 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:05.410012 write(1, " 4 S S S S . . . . . .   4 . . ."..., 48) = 48
1674  17:22:05.410410 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:05.410729 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:22:05.411047 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:05.411331 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:05.411730 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:05.412163 write(1, "\n", 1) = 1
1674  17:22:05.412527 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:05.412950 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:05.413386 read(0, "4 1\n", 1024) = 4
1674  17:22:08.080865 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:08.081239 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:08.083389 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:08.087051 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:08.509651 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:08.510595 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:08.511096 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:22:08.511558 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:08.513634 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:08.514825 write(1, "\n", 1) = 1
1674  17:22:08.515344 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:08.515849 write(1, "\n", 1) = 1
1674  17:22:08.516262 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:08.516752 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:08.517169 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:08.517587 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:08.518032 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:08.518347 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:08.518661 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:08.518954 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:08.519398 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:22:08.519757 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:08.520039 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:08.520343 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:08.520623 write(1, "\n", 1) = 1
1674  17:22:08.520850 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:08.521248 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:08.521580 read(0, "5 6\n", 1024) = 4
1674  17:22:11.232475 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:11.232935 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:11.235632 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:11.239726 write(1, "...Waiting for server to process"..., 41) = 41
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
```

```
1674  17:22:11.538062 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:11.538732 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:11.538985 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:22:11.539264 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:11.541038 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:11.542097 write(1, "\n", 1) = 1
1674  17:22:11.542503 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:11.542833 write(1, "\n", 1) = 1
1674  17:22:11.543214 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:11.543532 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:11.543851 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:11.544154 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:11.544452 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:11.544726 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:11.544973 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:11.545222 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:11.545458 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:22:11.545748 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:11.546066 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:11.546354 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:11.546653 write(1, "\n", 1) = 1
1674  17:22:11.546979 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:11.547320 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:11.547572 read(0, "5 7\n", 1024) = 4
1674  17:22:13.458658 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:13.459077 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:13.461724 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:13.463801 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:13.566178 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:13.566740 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:13.567046 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:22:13.567415 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:13.569231 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:13.570403 write(1, "\n", 1) = 1
1674  17:22:13.570739 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:13.571051 write(1, "\n", 1) = 1
1674  17:22:13.571358 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:13.571627 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:13.572044 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:13.572437 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:13.572781 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:13.573152 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:13.573435 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:13.573709 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:13.573955 write(1, " 6 . S . S . . . . . .   6 . . ."..., 48) = 48
1674  17:22:13.574182 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:13.574451 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:13.574716 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
```

```
1674  17:22:13.575066 write(1, "\n", 1) = 1
1674  17:22:13.575336 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:13.575614 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:13.575879 read(0, "6 \n", 1024) = 3
1674  17:22:16.496183 read(0, "1\n", 1024) = 2
1674  17:22:18.098781 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:18.099488 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:18.102169 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:18.104279 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:18.152230 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:18.153180 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:18.153385 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:22:18.153638 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:18.155348 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:18.158542 write(1, "\n", 1) = 1
1674  17:22:18.159245 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:18.159920 write(1, "\n", 1) = 1
1674  17:22:18.160295 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:18.160632 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:18.161012 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:18.161327 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:18.161614 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:18.161930 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:18.162248 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:18.162627 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:18.162868 write(1, " 6 . S . S . . . . . .   6 . X ."..., 48) = 48
1674  17:22:18.163118 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:18.163330 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:18.163529 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:18.163862 write(1, "\n", 1) = 1
1674  17:22:18.164152 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:18.164444 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:18.164718 read(0, "6 2\n", 1024) = 4
1674  17:22:21.036839 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:21.037157 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:21.039935 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:21.043341 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:21.398996 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:21.400002 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:21.400319 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:22:21.400669 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:21.403403 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:21.404452 write(1, "\n", 1) = 1
1674  17:22:21.404887 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:21.405277 write(1, "\n", 1) = 1
```

```
1674  17:22:21.405594 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:21.405917 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:21.406189 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:21.406549 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:21.406987 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:21.407322 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:21.407669 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:21.408013 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:21.408333 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
1674  17:22:21.408690 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:21.409068 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:21.409507 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:21.409862 write(1, "\n", 1) = 1
1674  17:22:21.410228 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:21.410627 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:21.411014 read(0, "6 3\n", 1024) = 4
1674  17:22:23.285405 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:23.285855 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:23.287912 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:23.289900 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:23.424476 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:23.425468 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:23.425749 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:22:23.426207 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:23.428192 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:23.430646 write(1, "\n", 1) = 1
1674  17:22:23.431211 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:23.431818 write(1, "\n", 1) = 1
1674  17:22:23.432414 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:23.432882 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:23.433225 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:23.433612 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:23.433896 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:23.434233 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:23.434579 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:23.434999 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:23.435410 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
1674  17:22:23.435738 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:23.436026 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:23.436364 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:23.436642 write(1, "\n", 1) = 1
1674  17:22:23.436980 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:23.437349 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:23.437672 read(0, "7 8\n", 1024) = 4
1674  17:22:26.067841 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:26.068343 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:26.071208 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:26.075055 write(1, "...Waiting for server to process"..., 41) = 41
```

```
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
1674  17:22:26.469695 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:26.470510 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:26.471013 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:22:26.471402 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:26.474819 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:26.476623 write(1, "\n", 1) = 1
1674  17:22:26.477298 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:26.477717 write(1, "\n", 1) = 1
1674  17:22:26.478074 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:26.478481 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:26.478829 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:26.479197 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:26.479589 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:26.480064 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:26.480569 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:26.480983 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:26.481327 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
1674  17:22:26.481675 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:26.482029 write(1, " 8 S S S . . . S S S .   8 . . ."..., 48) = 48
1674  17:22:26.482482 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:26.482867 write(1, "\n", 1) = 1
1674  17:22:26.483231 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:26.483627 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:26.484008 read(0, "8 0\n", 1024) = 4
1674  17:22:29.083250 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:29.083797 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:29.086509 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:29.088611 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:29.526579 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:29.527522 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:29.527947 write(1, "SUNK! You shoot again!\n", 23) = 23
1674  17:22:29.528347 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:29.530610 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:29.531309 write(1, "\n", 1) = 1
1674  17:22:29.531784 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:29.532236 write(1, "\n", 1) = 1
1674  17:22:29.532664 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:29.532939 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:29.533315 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:29.533636 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:29.533905 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:29.534204 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:29.534621 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:29.534994 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:29.535462 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
```

```
1674  17:22:29.535806 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:29.536027 write(1, " 8 S S S . . . S S S .   8 X . ."..., 48) = 48
1674  17:22:29.536323 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:29.536696 write(1, "\n", 1) = 1
1674  17:22:29.537009 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:29.537324 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:29.537641 read(0, "9 5\n", 1024) = 4
1674  17:22:34.738858 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:34.739472 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:34.742386 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:34.745890 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:35.167875 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:35.168648 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:35.168931 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:22:35.169246 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:35.171104 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:35.172197 write(1, "\n", 1) = 1
1674  17:22:35.172541 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:35.172847 write(1, "\n", 1) = 1
1674  17:22:35.173245 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:35.173695 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:35.174173 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:35.174669 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:35.175196 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:35.175671 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:35.176151 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:35.176602 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:35.177013 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
1674  17:22:35.177355 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:35.177727 write(1, " 8 S S S . . . S S S .   8 X . ."..., 48) = 48
1674  17:22:35.178046 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:35.178324 write(1, "\n", 1) = 1
1674  17:22:35.178583 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:35.178938 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:35.179223 read(0, "9 5\n", 1024) = 4
1674  17:22:36.217772 write(1, "Error: you already shot here!\n", 30) = 30
1674  17:22:36.218194 write(1, "\n", 1) = 1
1674  17:22:36.218551 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:36.218879 write(1, "\n", 1) = 1
1674  17:22:36.219282 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:36.219751 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:36.220092 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:36.220538 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:36.221015 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:36.221402 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:36.221842 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:36.222229 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:36.222600 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
```

```
1674  17:22:36.222927 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:36.223283 write(1, " 8 S S S . . . S S S .   8 X . ."..., 48) = 48
1674  17:22:36.223646 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:36.223977 write(1, "\n", 1) = 1
1674  17:22:36.224332 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:36.224583 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:36.224943 read(0, "9 4\n", 1024) = 4
1674  17:22:38.519949 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:38.520430 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:38.523235 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:38.525860 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:38.754805 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = -1 EAGAIN
    (Resource temporarily unavailable)
1674  17:22:38.755253 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:38.755656 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:22:38.756007 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:38.757856 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:38.758757 write(1, "\n", 1) = 1
1674  17:22:38.759087 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:38.759388 write(1, "\n", 1) = 1
1674  17:22:38.759620 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:38.759878 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:38.760074 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:38.760525 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:38.761034 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:38.761440 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:38.761728 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:38.762012 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:38.762335 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
1674  17:22:38.762672 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:38.762998 write(1, " 8 S S S . . . S S S .   8 X . ."..., 48) = 48
1674  17:22:38.763290 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:38.763577 write(1, "\n", 1) = 1
1674  17:22:38.763957 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:38.764346 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:38.764692 read(0, "9 6\n", 1024) = 4
1674  17:22:40.385710 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:40.386154 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:40.388490 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:40.390299 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:40.809380 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = -1 EAGAIN
    (Resource temporarily unavailable)
1674  17:22:40.809718 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:40.809943 write(1, "HIT! You shoot again!\n", 22) = 22
1674  17:22:40.810180 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
```

```
1674  17:22:40.813172 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:40.814164 write(1, "\n", 1) = 1
1674  17:22:40.814456 write(1, "========== CURRENT STATUS ======"..., 37) = 37
1674  17:22:40.814814 write(1, "\n", 1) = 1
1674  17:22:40.815128 write(1, "Your board:              Your sh"..., 37) = 37
1674  17:22:40.815446 write(1, "   0 1 2 3 4 5 6 7 8 9    0 1 2 "..., 46) = 46
1674  17:22:40.815746 write(1, " 0 . S . . . . . S S .   0 . . X"..., 48) = 48
1674  17:22:40.816025 write(1, " 1 . . . . . . . . . .   1 . . ."..., 48) = 48
1674  17:22:40.816319 write(1, " 2 S . . . . . S S . .   2 . . ."..., 48) = 48
1674  17:22:40.816626 write(1, " 3 . . . . . . . . . .   3 . X ."..., 48) = 48
1674  17:22:40.816993 write(1, " 4 S S S S . . . . . .   4 . X ."..., 48) = 48
1674  17:22:40.817296 write(1, " 5 . . . . . . S S . .   5 . . ."..., 48) = 48
1674  17:22:40.817538 write(1, " 6 . S . S . . . . . .   6 . X X"..., 48) = 48
1674  17:22:40.817792 write(1, " 7 . . . . . . . . . .   7 . . ."..., 48) = 48
1674  17:22:40.818094 write(1, " 8 S S S . . . S S S .   8 X . ."..., 48) = 48
1674  17:22:40.818466 write(1, " 9 . . . . . . . . . .   9 . . ."..., 48) = 48
1674  17:22:40.818878 write(1, "\n", 1) = 1
1674  17:22:40.819174 write(1, "=== YOUR TURN ===\n", 18) = 18
1674  17:22:40.819451 write(1, "Enter shot coordinates (row colu"..., 37) = 37
1674  17:22:40.819732 read(0, "9 7\n", 1024) = 4
1674  17:22:42.390015 write(1, "...Sending shot to server...\n", 29) = 29
1674  17:22:42.390579 msync(0x7f7fa8a9c000, 2048, MS_SYNC) = 0
1674  17:22:42.393227 msync(0x7f7fa8a9c000, 32864, MS_SYNC) = 0
1674  17:22:42.395205 write(1, "...Waiting for server to process"..., 41) = 41
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
1674  17:22:42.839394 futex(0x7f7fa8a9c7a8, FUTEX_WAIT, 2, NULL) = 0
1674  17:22:42.847403 futex(0x7f7fa8a9c7a8, FUTEX_WAKE, 1) = 0
1674  17:22:42.847635 write(1, "*** YOU WON! ***\n", 17) = 17
1674  17:22:42.847880 munmap(0x7f7fa8a9c000, 32864) = 0
1674  17:22:42.848067 close(3)          = 0
1674  17:22:42.848226 openat(AT_FDCWD, "players_stats.db", O_RDONLY) = 3
1674  17:22:42.848427 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=76004,
    ...}, AT_EMPTY_PATH) = 0
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"..., 73728) =
    73728
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
1674  17:22:42.849159 close(3)          = 0
1674  17:22:42.849401 write(1, "\n", 1) = 1
1674  17:22:42.849610 write(1, "Statistics updated from server\n", 31) = 31
1674  17:22:42.849886 write(1, "\n", 1) = 1
1674  17:22:42.850108 write(1, "========== MAIN MENU ==========\n", 32) = 32
1674  17:22:42.850304 write(1, "1. Create new game\n", 19) = 19
1674  17:22:42.850485 write(1, "2. Join a game\n", 15) = 15
1674  17:22:42.850774 write(1, "3. View my statistics\n", 22) = 22
1674  17:22:42.851139 write(1, "4. Exit\n", 8) = 8
1674  17:22:42.851349 write(1, "Choose action (1-4): ", 21) = 21
```

```
1674  17:22:42.851534 read(0, "3\n", 1024) = 2
1674  17:22:44.309492 openat(AT_FDCWD, "players_stats.db", O_RDONLY) = 3
1674  17:22:44.309779 newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=76004,
    ...}, AT_EMPTY_PATH) = 0
1674  17:22:44.310108 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"..., 73728) =
    73728
1674  17:22:44.310506 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
1674  17:22:44.310978 close(3)          = 0
1674  17:22:44.311517 write(1, "\n", 1) = 1
1674  17:22:44.312100 write(1, "========== PLAYER STATISTICS ==="..., 40) = 40
1674  17:22:44.312552 write(1, "Login: kolko\n", 13) = 13
1674  17:22:44.312948 write(1, "Wins: 5\n", 8) = 8
1674  17:22:44.313431 write(1, "Loses: 0\n", 9) = 9
1674  17:22:44.313847 write(1, "Total games: 5\n", 15) = 15
1674  17:22:44.314325 write(1, "Win rate: 100.0%\n", 17) = 17
1674  17:22:44.314846 write(1, "\n", 1) = 1
1674  17:22:44.315263 write(1, "========== MAIN MENU ==========\n", 32) = 32
1674  17:22:44.315671 write(1, "1. Create new game\n", 19) = 19
1674  17:22:44.316114 write(1, "2. Join a game\n", 15) = 15
1674  17:22:44.316486 write(1, "3. View my statistics\n", 22) = 22
1674  17:22:44.317052 write(1, "4. Exit\n", 8) = 8
1674  17:22:44.317668 write(1, "Choose action (1-4): ", 21) = 21
1674  17:22:44.318071 read(0, "4\n", 1024) = 2
1674  17:22:46.223892 write(1, "Goodbye!\n", 9) = 9
1674  17:22:46.224372 exit_group(0)     = ?
1674  17:22:46.224837 +++ exited with 0 +++
```

Листинг 16: *Strace логи клиента*