Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Курсовой проект по курсу

«Операционные системы»

Группа: М80-207Б-20

Студент: Мерц С.П.

Вариант: 7

Преподаватель: Миронов Е.С.

Оценка: ____________________

Дата: 27.12.21

Москва, 2021.

Содержание

Постановка задачи

Проектирование консольной клиент-серверной игры На основе любой из выбранных технологий: Pipes Sockets Сервера очередей И другие Создать собственную игру более, чем для одного пользователя. Игра может быть устроена по принципу: клиент-клиент, сервер-клиент.

Общие сведения о программе

Программа состоит из двух файлов, которые компилируются в исполнительные файлы(которые представляют управляющие и вычислительные узлы). Общение между процессами происходит с помощью именованных пайпов.

Общий метод и алгоритм решения

· Клиент отправляет запросы и принимает ответы в отдельном потоке.

· Сервер отвечает на запросы клиента и запускает игры в отдельных процессах.

Код программы

server.cpp:

```
 #include <iostream>

#include <stdlib.h>
#include <unistd.h>
#include <sys/stat.h>
#include <string>
#include <sys/types.h>
#include <errno.h>
#include <vector>
#include <fcntl.h>
#include <map>
#include <vector>
#include <thread>
#include "funcs.hpp"

#define CLIENT_ID(name) in(logins,name)
#define PLAYER_ID(name) in(curr_playrs_name, name)

inline int create_game_pipe(std::string game_name){
    int curr_playrs;
    if (mkfifo(("game_%" + game_name).c_str(), S_IRWXU|S_IRWXG|S_IRWXO) == -1)
    {
        std::cout << "GAME " << ("game_%" + game_name).c_str() << " FIFO WAS NOT CREATED";
        exit(1);
    }
    int game_input_fd = open(("game_%" + game_name).c_str(), O_RDWR);
    if (game_input_fd == -1)
    {
        std::cout << "MAIN INPUT FIFO WAS NOT OPENED";
        exit(1);
    }
    return game_input_fd;
}

inline int in(std::vector<std::string> logins, std::string str)
{
    for (int i = 0; i < logins.size(); ++i)
    {
        if (logins[i] == str)
            return i;
    }
    return -1;
}

inline int create_main_pipe() {
    if (mkfifo("main_input", S_IRWXU|S_IRWXG|S_IRWXO) == -1)
    {
        std::cout << "MAIN INPUT FIFO WAS NOT CREATED";
        exit(1);
    }
    int fd_recv = open("main_input", O_RDWR);
    if (fd_recv == -1)
    {
        std::cout << "MAIN INPUT FIFO WAS NOT OPENED";
        exit(1);
    }
    return fd_recv;
}

inline int create_admin_pipe() {
    if (mkfifo("admin", S_IRWXU|S_IRWXG|S_IRWXO) == -1)
    {
        std::cout << "ADMIN INPUT FIFO WAS NOT CREATED";
        exit(1);
    }
    int admin_fd = open("admin", O_RDWR);
    if (admin_fd == -1)
    {
        std::cout << "ADMIN INPUT FIFO WAS NOT OPENED";
        exit(1);
    }
    return admin_fd;
}

inline int create_client_pipe(std::string rcvd_name) {
    if (mkfifo(rcvd_name.c_str(), S_IRWXU|S_IRWXG|S_IRWXO) == -1)
    {
        std::cout << "CLIENT INPUT FIFO WAS NOT CREATED";
        exit(1);
    }
    int fd = open(rcvd_name.c_str(), O_RDWR);
    if (fd == -1)
    {
        std::cout << "CLIENT INPUT FIFO WAS NOT OPENED";
        exit(1);
    }
    return fd;
}

int hit_check (std::string game_word, std::string try_word, int *cows, int *bulls) {
    if (try_word.size() != game_word.size())
        return -1;
    if (try_word == game_word)
        return -2;

    *cows = 0;
    *bulls = 0;

    for (size_t i = 0; i < try_word.size(); ++i)
        for (size_t j = 0; j < game_word.size(); ++j)
            if (try_word[i] == game_word[j])
                *cows = *cows + 1;
    for (size_t i = 0; i < try_word.size(); ++i)
        if (try_word[i] == game_word[i])
            *bulls = *bulls + 1;
    return 0;
}

void game_funk (std::string game_name, std::string game_word)
{
    std::vector<std::string> curr_playrs_name;
    std::vector<int> curr_playrs_fd;
    auto iter_fd = curr_playrs_fd.cbegin();
    auto iter_log = curr_playrs_name.cbegin();
    int game_input_fd = create_game_pipe(game_name);
    int cows, bulls;
    std::string game_respond;
    int game_status;

    std::cout << "START GAME: " << game_name << std::endl;
    std::cout.flush();

    std::string rcvd_name, rcvd_command, rcvd_data;
    while (1)
    {
        recieve_message_server(game_input_fd, &rcvd_name, &rcvd_command, &rcvd_data);
        if (rcvd_command == "connect")
        {
            curr_playrs_name.push_back(rcvd_name);
            curr_playrs_fd.push_back(open(rcvd_name.c_str(), O_RDWR));

            std::cout << "CLIENT: " << rcvd_name << " JOIN GAME: " << game_name << std::endl;
            std::cout.flush();

            game_respond = "Добро пожаловать за стол " + game_name;
            send_message_to_client(curr_playrs_fd[PLAYER_ID(rcvd_name)], game_respond.c_str());
            game_respond = "Делайте свои предпололения с помощью команды maybe @слово@";
            send_message_to_client(curr_playrs_fd[PLAYER_ID(rcvd_name)], game_respond.c_str());
        }
        else if (rcvd_command == "maybe")
        {
            game_status = hit_check(game_word, rcvd_data, &cows, &bulls);

            if (game_status == -1)
            {
                game_respond = "Размеры слов не совподают";
                send_message_to_client(curr_playrs_fd[PLAYER_ID(rcvd_name)], game_respond.c_str());
            }
            else if (game_status == -2)
            {
                game_respond = "Вы выйграли";
                send_message_to_client(curr_playrs_fd[PLAYER_ID(rcvd_name)], game_respond.c_str());

                for (int i=0; i < curr_playrs_name.size(); i++)
                {
                    game_respond = "Иргру выйграл: " + rcvd_name + "\nЗагаданное слово: " + game_word;
                    send_message_to_client(curr_playrs_fd[i], game_respond.c_str());
                    do{
                        game_respond = "Выйдите из-за стола (команда leave)";
                        send_message_to_client(curr_playrs_fd[i], game_respond.c_str());
                        recieve_message_server(game_input_fd, &rcvd_name, &rcvd_command, &rcvd_data);
                    }while(rcvd_command != "leave");
                }

                close(game_input_fd);

                std::cout<<"TEST\n";
                std::cout.flush();

                std::cout << "FINISH GAME: " << game_name << std::endl;
                std::cout.flush();
                int mainFD = open("main_input", O_RDWR);
                game_respond = "finish";
                send_message_to_server(mainFD, game_name, game_respond, "");
                std::cout<<"TEST\n";
                std::cout.flush();
                return;
            }
            else if (game_status == 0)
            {
                game_respond = "Коровы: " + std::to_string(cows) + " Быки: " + std::to_string(bulls);
                send_message_to_client(curr_playrs_fd[PLAYER_ID(rcvd_name)], game_respond.c_str());
            }
        }
        else if (rcvd_command == "leave")
        {
            iter_fd = curr_playrs_fd.cbegin();
            curr_playrs_fd.erase(iter_fd + PLAYER_ID(rcvd_name));
            iter_log = curr_playrs_name.cbegin();
            curr_playrs_name.erase(iter_log + PLAYER_ID(rcvd_name));
            std::cout << "CLIENT: " << rcvd_name << " LEFT GAME: " << game_name << std::endl;
        }
    }
}

int main()
{
    std::vector<std::string> logins;
    std::vector<int> client_pipe_fd;
    std::vector<std::thread> games_threads;
    std::vector<std::string> games_name;
    std::string game_name_table, game_word;

    int fd_recv = create_main_pipe();
    int admin_fd = create_admin_pipe();

    std::string login;
    std::string rcvd_name, rcvd_command, rcvd_data;
    auto iter_fd = client_pipe_fd.cbegin();
    auto iter_log = logins.cbegin();
    auto iter_game_thread = games_threads.cbegin();
    auto iter_game_name = games_name.cbegin();
    while (1)
    {
        recieve_message_server(fd_recv, &rcvd_name, &rcvd_command, &rcvd_data);

        if (rcvd_command == "login" && rcvd_name != "admin")
        {
            std::cout <<"New client: " << rcvd_name << std::endl;
            std::cout.flush();

            client_pipe_fd.push_back(create_client_pipe(rcvd_name));
            logins.push_back(rcvd_name);
        }
        else if (rcvd_command == "create")
        {
            extract_game_data(rcvd_data, &game_name_table, &game_word);
            games_name.push_back(game_name_table);
            games_threads.push_back(std::thread(game_funk, game_name_table, game_word));
        }
        else if (rcvd_command == "finish" /*&& rcvd_name.substr(1,6) == "game_%"*/)
        {
            std::remove(("game_%" + rcvd_name).c_str());
            std::cout<<"TEST\n";
                std::cout.flush();
            games_threads[in(games_name, rcvd_name)].detach();
            std::cout<<"TEST\n";
                std::cout.flush();
            iter_game_thread = games_threads.cbegin();
            games_threads.erase(iter_game_thread + in(games_name, rcvd_name));
            std::cout<<"TEST\n";
                std::cout.flush();
            iter_game_name = games_name.cbegin();
            games_name.erase(iter_game_name + in(games_name, rcvd_name));
            std::cout<<"TEST\n";
                std::cout.flush();

        }
        else if (rcvd_command == "quit")
        {
            close(client_pipe_fd[CLIENT_ID(rcvd_name)]);
            std::remove(rcvd_name.c_str());

            iter_fd = client_pipe_fd.cbegin();
            iter_log = logins.cbegin();

            client_pipe_fd.erase(iter_fd + CLIENT_ID(rcvd_name));
            logins.erase(iter_log + CLIENT_ID(rcvd_name));
            std::cout << "CLIENT: " << rcvd_name << " LEFT\n";

        }
        else if (rcvd_command == "shut_down" && rcvd_name == "admin")
        {
            for(int i=0 ; i < logins.size(); i++)
            {
                send_message_to_client(client_pipe_fd[i],"SERVER CLOSED");
                close(client_pipe_fd[i]);
                std::remove(logins[i].c_str());
            }
            for(int i=0 ; i < games_threads.size(); i++)
```

```
            {
                std::remove(games_name[i].c_str());
                games_threads[i].detach();
            }

            close(admin_fd);
            std::remove("admin");
            std::remove("main_input");
            std::cout << "SERVER OFF\n";

            return 0;
        }
        else if(rcvd_name != "admin")
        {
            // send_message_to_client(client_pipe_fd[CLIENT_ID(rcvd_name)],"NOT A COMMAND");
        }
    }
}
```

client.cpp:

```
#include <iostream>
#include <stdlib.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <errno.h>
#include <vector>
#include <fcntl.h>
#include "funcs.hpp"
#include <thread>

#define SEND_TO_SERVER(FD) send_message_to_server(FD, login, command, data)

//функция приёма сообщений (для потока)
void func(int fd_respond, std::string login)
{
    while (1)
    {
        std::string respond = recieve_message_client(fd_respond);
        std::cout << "\n" << respond << "\n";
        std::cout.flush();
        if (respond == "SERVER CLOSED")
            exit(0);
        std::cout << login << "> ";
        std::cout.flush();
    }
}

inline void write_intro() {
    std::cout << "Добро пожаловать в игру Быки и Коровы.\nЧтобы создать аккаунт запустите ./server и введите там свой логи\n";
    std::cout <<  "Затем перезапустите клиент и впишите совй логин\n";
    std::cout << "Введите свой логин: ";
    std::cout.flush();
}

inline void write_menu(std::string login){
    std::cout << "Соединение установлено, можете отдавать команды\n";
    std::cout << "Список команд:\n";
    std::cout << "1) create @название игрового стола@ @игровое слово@\n";
    std::cout << "2) connect @название игры@\n";
    std::cout << "3) leave\n";
    if (login != "admin")std::cout << "4) quit\n";
    if (login == "admin") std::cout << "5) shut_down - выключение сервера\n";
    std::cout.flush();
}

inline int server_main_input_fd(){

    int fd = open("main_input", O_RDWR);
    if (fd == -1)
    {
        std::cout << "ERROR: MAIN FIFO WAS NOT OPENED\n";
        exit(1);
    }
    return fd;
}

int main()
{
    int client_main_out_fd = server_main_input_fd();

    write_intro();
    std::string login;
    std::cin >> login;
    send_message_to_server(client_main_out_fd, login, "login", "");
    std::cout << "Устанавливаю соединение\n";
    sleep(1);
    int fd_respond = open(login.c_str(), O_RDWR);
    if (fd_respond == -1)
    {
        std::cout << "RESPOND FIFO WAS NOT OPENED";
        exit(1);
    }

    write_menu(login);
    std::thread thr_respond(func, fd_respond, login);

    std::string command, data;
    std::string game_word, game_name;
    int game_fd;

    while (1)
    {
        std::cout << login << "> ";
        std::cin >> command;

        if (command == "create")
        {
            std::cin >> game_name >> game_word;
            data = game_name + "$" + game_word;
            SEND_TO_SERVER(client_main_out_fd);
        }
        else if (command == "connect")
        {
            std::cin >> game_name;
            game_fd = open(("game_%" + game_name).c_str(), O_RDWR);
            if (game_fd == -1)
            {
                std::cout << "ERROR: GAME NOT FOUND\n";
                std::cout.flush();
            }
            else
            {
                data = "";
                SEND_TO_SERVER(game_fd);
                std::cout << login << "> ";
                std::cout.flush();
                while (1)
                {
                    std::cin >> command;

                    if (command == "maybe")
                    {
                        std::cin >> data;
                        SEND_TO_SERVER(game_fd);
                    }
                    else if (command == "leave")
                    {
                        data = "";
                        SEND_TO_SERVER(game_fd);
                        break;
                    }
                    else
                    {
                        std::cout << login << "> ";
                        std::cout.flush();
                    }
                }
            }
        }
        else if (command == "quit" && login != "admin")
        {
            data = "";
            SEND_TO_SERVER(client_main_out_fd);
            thr_respond.detach();
            return 0;
        }
        else if (command == "shut_down" && login == "admin")
        {
            data = "";
            SEND_TO_SERVER(client_main_out_fd);
            thr_respond.detach();
            return 0;
        }
    }
    return 0;
}
```

STRACE

./server

```
execve("./server", ["./server"], 0x7ffeb9286330 /* 64 vars */) = 0

brk(NULL)                               = 0x560da48d3000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffffd3eb190) = -1 EINVAL (Invalid argument)
```

```
access("/etc/ld.so.preload", R_OK)    = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=130548, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 130548, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ff5333f8000

close(3)                                = 0

openat(AT_FDCWD, "/usr/lib/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\220\t\0\0\0\0\0"..., 832) = 832

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\212\260\345pT\335\35\313\246\201\362\27\1j\374j"..., 36,
800) = 36

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=17969672, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7ff5333f6000

mmap(NULL, 2185280, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff5331e0000

mmap(0x7ff533279000, 1048576, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x99000) = 0x7ff533279000

mmap(0x7ff533379000, 442368, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x199000) = 0x7ff533379000

mmap(0x7ff5333e5000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x204000) = 0x7ff5333e5000

mmap(0x7ff5333f3000, 10304, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ff5333f3000

close(3)                                = 0

openat(AT_FDCWD, "/usr/lib/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260\363\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1323472, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 1323032, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff53309c000

mprotect(0x7ff5330ab000, 1257472, PROT_NONE) = 0

mmap(0x7ff5330ab000, 630784, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0x7ff5330ab000

mmap(0x7ff533145000, 622592, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0xa9000) = 0x7ff533145000

mmap(0x7ff5331de000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x141000) = 0x7ff5331de000

close(3)                                = 0
```

```
openat(AT_FDCWD, "/usr/lib/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=475944, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 107240, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff533081000

mprotect(0x7ff533084000, 90112, PROT_NONE) = 0

mmap(0x7ff533084000, 73728, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7ff533084000

mmap(0x7ff533096000, 12288, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x15000) = 0x7ff533096000

mmap(0x7ff53309a000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7ff53309a000

close(3)                                = 0

openat(AT_FDCWD, "/usr/lib/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300\200\0\0\0\0\0\0"..., 832) = 832

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"..., 80, 792) = 80

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\310\371[O2Q\320\205P!z\330\241\363\20"..., 68, 872) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=154040, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 131472, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff533060000

mprotect(0x7ff533067000, 81920, PROT_NONE) = 0

mmap(0x7ff533067000, 61440, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7ff533067000

mmap(0x7ff533076000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x16000) = 0x7ff533076000

mmap(0x7ff53307b000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7ff53307b000

mmap(0x7ff53307d000, 12688, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ff53307d000

close(3)                                = 0

openat(AT_FDCWD, "/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`|\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

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"..., 80, 848) = 80

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0K@g7\5w\10\300\344\306B4Zp<G"..., 68, 928) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2150424, ...}, AT_EMPTY_PATH) = 0
```

```
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 1880536, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff532e94000

mmap(0x7ff532eba000, 1355776, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x7ff532eba000

mmap(0x7ff533005000, 311296, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x171000) = 0x7ff533005000

mmap(0x7ff533051000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bc000) = 0x7ff533051000

mmap(0x7ff533057000, 33240, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ff533057000

close(3)                                = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7ff532e92000

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7ff532e8f000

arch_prctl(ARCH_SET_FS, 0x7ff532e8f740) = 0

mprotect(0x7ff533051000, 12288, PROT_READ) = 0

mprotect(0x7ff53307b000, 4096, PROT_READ) = 0

mprotect(0x7ff53309a000, 4096, PROT_READ) = 0

mprotect(0x7ff5331de000, 4096, PROT_READ) = 0

mprotect(0x7ff5333e5000, 53248, PROT_READ) = 0

mprotect(0x560da3abb000, 4096, PROT_READ) = 0

mprotect(0x7ff533446000, 8192, PROT_READ) = 0

munmap(0x7ff5333f8000, 130548)          = 0

set_tid_address(0x7ff532e8fa10)         = 2014

set_robust_list(0x7ff532e8fa20, 24)     = 0

rt_sigaction(SIGRTMIN, {sa_handler=0x7ff533067b70, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7ff533073870}, NULL, 8) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x7ff533067c10, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7ff533073870}, NULL, 8) = 0

rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

brk(NULL)                               = 0x560da48d3000

brk(0x560da48f4000)                     = 0x560da48f4000
```

```
futex(0x7ff5333f36bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x7ff5333f36c8, FUTEX_WAKE_PRIVATE, 2147483647) = 0
mknodat(AT_FDCWD, "main_input", S_IFIFO|0777) = 0
openat(AT_FDCWD, "main_input", O_RDWR)  = 3
mknodat(AT_FDCWD, "admin", S_IFIFO|0777) = 0
openat(AT_FDCWD, "admin", O_RDWR)   = 4
read(3, "\n\0\0\0", 4)              = 4
read(3, "man$login$", 10)                  = 10
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
write(1, "New client: man\n", 16New client: man
)           = 16
mknodat(AT_FDCWD, "man", S_IFIFO|0777)  = 0
openat(AT_FDCWD, "man", O_RDWR)       = 5
read(3, "\t\0\0\0", 4)              = 4
read(3, "man$quit$", 9)                    = 9
close(5)                            = 0
unlink("man")                       = 0
write(1, "CLIENT: man LEFT\n", 17CLIENT: man LEFT
)           = 17
read(3, "\f\0\0\0", 4)              = 4
read(3, "admin$login$", 12)                = 12
read(3, "\20\0\0\0", 4)                    = 4
read(3, "admin$shut_down$", 16)     = 16
close(4)                            = 0
unlink("admin")                     = 0
unlink("main_input")                       = 0
write(1, "SERVER OFF\n", 11SERVER OFF
)           = 11
exit_group(0)                       = ?
+++ exited with 0 +++
```

./client

```
execve("./client", ["./client"], 0x7fff04891cf0 /* 64 vars */) = 0

brk(NULL)                        = 0x55a74d6e4000

arch_prctl(0x3001 /* ARCH_??? */, 0x7fff9729ff50) = -1 EINVAL (Invalid argument)

access("/etc/ld.so.preload", R_OK)    = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=130548, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 130548, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f409f0cb000

close(3)                         = 0

openat(AT_FDCWD, "/usr/lib/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\220\t\0\0\0\0\0"..., 832) = 832

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\212\260\345pT\335\35\313\246\201\362\27\1j\374j"..., 36,
800) = 36

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=17969672, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f409f0c9000

mmap(NULL, 2185280, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f409eeb3000

mmap(0x7f409ef4c000, 1048576, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x99000) = 0x7f409ef4c000

mmap(0x7f409f04c000, 442368, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x199000) = 0x7f409f04c000

mmap(0x7f409f0b8000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x204000) = 0x7f409f0b8000

mmap(0x7f409f0c6000, 10304, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f409f0c6000

close(3)                         = 0

openat(AT_FDCWD, "/usr/lib/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260\363\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1323472, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 1323032, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f409ed6f000

mprotect(0x7f409ed7e000, 1257472, PROT_NONE) = 0
```

```
mmap(0x7f409ed7e000, 630784, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0x7f409ed7e000

mmap(0x7f409ee18000, 622592, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0xa9000) = 0x7f409ee18000

mmap(0x7f409eeb1000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x141000) = 0x7f409eeb1000

close(3)                                = 0

openat(AT_FDCWD, "/usr/lib/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=475944, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 107240, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f409ed54000

mprotect(0x7f409ed57000, 90112, PROT_NONE) = 0

mmap(0x7f409ed57000, 73728, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f409ed57000

mmap(0x7f409ed69000, 12288, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x15000) = 0x7f409ed69000

mmap(0x7f409ed6d000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7f409ed6d000

close(3)                                = 0

openat(AT_FDCWD, "/usr/lib/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300\200\0\0\0\0\0\0"..., 832) = 832

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"..., 80, 792) = 80

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\310\371[O2Q\320\205P!z\330\241\363\20"..., 68, 872) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=154040, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 131472, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f409ed33000

mprotect(0x7f409ed3a000, 81920, PROT_NONE) = 0

mmap(0x7f409ed3a000, 61440, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f409ed3a000

mmap(0x7f409ed49000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x16000) = 0x7f409ed49000

mmap(0x7f409ed4e000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7f409ed4e000

mmap(0x7f409ed50000, 12688, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f409ed50000

close(3)                                = 0
```

```
openat(AT_FDCWD, "/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`|\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

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"..., 80, 848) = 80

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0K@g7\5w\10\300\344\306B4Zp<G"..., 68, 928) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2150424, ...}, AT_EMPTY_PATH) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 1880536, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f409eb67000

mmap(0x7f409eb8d000, 1355776, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x7f409eb8d000

mmap(0x7f409ecd8000, 311296, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x171000) = 0x7f409ecd8000

mmap(0x7f409ed24000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bc000) = 0x7f409ed24000

mmap(0x7f409ed2a000, 33240, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f409ed2a000

close(3)                                = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f409eb65000

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f409eb62000

arch_prctl(ARCH_SET_FS, 0x7f409eb62740) = 0

mprotect(0x7f409ed24000, 12288, PROT_READ) = 0

mprotect(0x7f409ed4e000, 4096, PROT_READ) = 0

mprotect(0x7f409ed6d000, 4096, PROT_READ) = 0

mprotect(0x7f409eeb1000, 4096, PROT_READ) = 0

mprotect(0x7f409f0b8000, 53248, PROT_READ) = 0

mprotect(0x55a74c620000, 4096, PROT_READ) = 0

mprotect(0x7f409f119000, 8192, PROT_READ) = 0

munmap(0x7f409f0cb000, 130548)          = 0

set_tid_address(0x7f409eb62a10)         = 2072

set_robust_list(0x7f409eb62a20, 24) = 0

rt_sigaction(SIGRTMIN, {sa_handler=0x7f409ed3ab70, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f409ed46870}, NULL, 8) = 0
```

```
rt_sigaction(SIGRT_1, {sa_handler=0x7f409ed3ac10, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f409ed46870}, NULL, 8) = 0

rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

brk(NULL)                       = 0x55a74d6e4000

brk(0x55a74d705000)             = 0x55a74d705000

futex(0x7f409f0c66bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0

futex(0x7f409f0c66c8, FUTEX_WAKE_PRIVATE, 2147483647) = 0

openat(AT_FDCWD, "main_input", O_RDWR)  = 3

newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH) = 0

write(1, "\320\224\320\276\320\261\321\200\320\276
\320\277\320\276\320\266\320\260\320\273\320\276\320\262\320\260\321\202\321\214 "..., 70Добро
пожаловать в игру Быки и Коровы.

) = 70

write(1, "\320\247\321\202\320\276\320\261\321\213
\321\201\320\276\320\267\320\264\320\260\321\202\321\214 \320\260\320\272\320\272"..., 112Чтобы
создать аккаунт запустите ./server и введите там свой логи

) = 112

write(1, "\320\227\320\260\321\202\320\265\320\274
\320\277\320\265\321\200\320\265\320\267\320\260\320\277\321\203\321\201\321\202\320"..., 89Затем
перезапустите клиент и впишите совй логин

) = 89

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\320\262\320\276\320\271 \320\273\320\276\320\263\320\270"..., 36Введите свой логин: ) = 36

newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH) = 0

read(0, man

"man\n", 1024)                  = 4

write(3, "\n\0\0\0", 4)         = 4

write(3, "man$login$", 10)              = 10

write(1,
"\320\243\321\201\321\202\320\260\320\275\320\260\320\262\320\273\320\270\320\262\320\260\321\21
6 \321\201\320\276\320\265\320"..., 46Устанавливаю соединение

) = 46

clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=1, tv_nsec=0}, 0x7fff9729fe50) = 0

openat(AT_FDCWD, "man", O_RDWR)       = 4
```

```
write(1, "\320\241\320\276\320\265\320\264\320\270\320\275\320\265\320\275\320\270\320\265
\321\203\321\201\321\202\320\260\320\275\320"..., 90Соединение установлено, можете отдавать
команды

) = 90

write(1, "\320\241\320\277\320\270\321\201\320\276\320\272
\320\272\320\276\320\274\320\260\320\275\320\264:\n", 27Список команд:

) = 27

write(1, "1) create @\320\275\320\260\320\267\320\262\320\260\320\275\320\270\320\265
\320\270\320\263"..., 851) create @название игрового стола@ @игровое слово@

) = 85

write(1, "2) connect @\320\275\320\260\320\267\320\262\320\260\320\275\320\270\320\265
\320\270\320"..., 392) connect @название игры@

) = 39

write(1, "3) leave\n", 93) leave

)                  = 9

write(1, "4) quit\n", 84) quit

)                  = 8

mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f409e361000

mprotect(0x7f409e362000, 8388608, PROT_READ|PROT_WRITE) = 0

rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0

clone(child_stack=0x7f409eb60ef0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYS
VSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[2075],
tls=0x7f409eb61640, child_tidptr=0x7f409eb61910) = 2075

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

write(1, "man> ", 5man> )               = 5

read(0, quit

"quit\n", 1024)                   = 5

write(3, "\t\0\0\0", 4)           = 4

write(3, "man$quit$", 9)                = 9

lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Illegal seek)

exit_group(0)                     = ?

+++ exited with 0 +++
```

Вывод

Про именованные пайпы я прочитал ещё во время выполнения второй лабораторной работы, и мне кажется то один из самых удобных способов

коммуникации между процессами. Также я писал серверное приложение уже не первый раз (второй), предыдущий опыт, полученный на летней практике очень помог в разработке и представлении всех связей между клиентом и сервером. Хоть мне и не особо нравиться веб разработка, но бэкэнд лучшее, что в ней есть и эта весьма интересное направления для меня как и в качестве развития своих навыков, так и в качестве первого опыта работы, ведь разработчики этого направления очень востребованы и требуемый уровень подготовки весьма низкий.