

Практикум на ЭВМ

Семестровая работа №3.

game-server

Баев А.Ж.

Казахстанский филиал МГУ

16 ноября 2018

1. Сервер

- ждет подключения N игроков
- рассылает всем игрокам стартовую карту
- обрабатывает ходы игроков

2. Клиент

- подключается к серверу
- ждет стартовую карту
- выполняет ходы

include

```
#include <unistd.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

Сервер. Создание сокета

```
//open socket, return socket descriptor  
int server_socket = socket(PF_INET, SOCK_STREAM, 0);  
  
//set socket option  
int socket_option = 1;  
setsockopt(server_socket,  
            SOL_SOCKET,  
            SO_REUSEADDR,  
            &socket_option,  
            sizeof(socket_option));
```

Сервер. Привязка адреса

```
//set socket address  
struct sockaddr_in server_address;  
server_address.sin_family = AF_INET;  
server_address.sin_port = htons(8080);  
server_address.sin_addr.s_addr = INADDR_ANY;  
bind(server_socket,  
      (struct sockaddr *) &server_address,  
      sizeof(server_address));  
  
//start listen mode  
listen(server_socket, 5);
```

```
while(1) {
    struct sockaddr_in client;
    socklen_t size;
    int client_socket;
    client_socket = accept(server_socket,
                           (struct sockaddr *) &client,
                           &size);

    char *addr = inet_ntoa(client.sin_addr);
    int port = ntohs(client.sin_port);

    printf("connected: %s %d\n", addr, port);

    //write(client_socket, ...)

    close(client_socket);
}
```

Клиент. Создание сокета

```
//open socket, return socket descriptor  
int server_socket = socket(PF_INET, SOCK_STREAM, 0);  
  
//prepare server address  
struct hostent *host = gethostbyname(ip);  
struct sockaddr_in server_address;  
server_address.sin_family = AF_INET;  
server_address.sin_port = htons(port);  
memcpy(&server_address.sin_addr,  
       host->h_addr_list[0],  
       sizeof(server_address));
```

```
//connect  
connect(server_address, (struct sockaddr *) &server_addr,  
  
//read(server_socket, ...)
```


Неканонический терминал (библиотека ncurses)

gcc prog.c -lcurses

```
#include < curses.h>
#include < locale.h>
int main(void) {
    int prev_x = 0, prev_y = 0, caret_y = 0, caret_x = 0;
    int flag = 1;
    setlocale(LC_ALL, "");
    if (!initscr()) return 1;
    cbreak();
    noecho();
    nonl();
    meta(stdscr, TRUE);
    intrflush(stdscr, FALSE);
    keypad(stdscr, TRUE);

    if (has_colors()) {
        start_color();
        init_pair(1, COLOR_WHITE, COLOR_BLUE);
    }
}
```

Неканонический терминал (библиотека ncurses)

```
attrset(COLOR_PAIR(1));  
bkgdset(COLOR_PAIR(1));  
clear();  
while (flag) {  
    mvaddch(prev_y, prev_x, '␣');  
    mvaddch(caret_y, caret_x, '*');  
    move(caret_y, caret_x);  
    refresh();  
    prev_x = caret_x;  
    prev_y = caret_y;  
}
```

Неканонический терминал (библиотека ncurses)

```
c = getch();
switch (c) {
case 033:
    flag = 0;
    break;
case KEY_UP:
    if (caret_y > 0) caret_y--;
    break;
case KEY_DOWN:
    if (caret_y < LINES - 1) caret_y++;
    break;
case KEY_LEFT:
    if (caret_x > 0) caret_x--;
    break;
case KEY_RIGHT:
    if (caret_x < COLS - 2) caret_x++;
    break;
}
```

Неканонический терминал (библиотека ncurses)

```
bkgdset(COLOR_PAIR(0));  
clear();  
refresh();  
endwin();  
return 0;  
}
```

Задание практикума: сетевая игра, консольный Doom

Алексей Сальников

- 1) Требуется реализовать игровой сервер, игровой клиент, отображатель статистики.
- 2) Сервер и клиенты должны взаимодействовать через сеть путём установки TCP соединения.
- 3) При своём запуске сервер читает файл с картой (имя файла указывается в параметрах при запуске).
- 4) Клиент соединится с сервером (hostname сервера и номер TCP порта сервера указываются клиенту в аргументах main при запуске)

Задание

Игра происходит в прямоугольном лабиринте, представленным как набор точек некоторой матрицы размера $M \times N$. Лабиринт вместе с его размером должны быть заданы в файле карты, при этом сам файл карты должен иметь текстовое представление легко читаемое человеком. Цель игры оказаться выжившим в лабиринте с максимальным уровнем здоровья.

10 20

```
#####
# #  #  #
# #  #  #####
# #####  #  #
#          #####
#  # ##          #
# #####  # #####
## ##          #
## ### #####
#  #          #  #
#          #  #
```

Лабиринт состоит из стенок, аптек/отравлялок, и проходов. По точке, содержащей аптеку/отравлялку можно двигаться, по стенкам и за границами лабиринта двигаться нельзя. Аптека обладает лечебным/отравляющим эффектом определённой силы. Игрок может съесть аптеку/отравлялку, при этом к его здоровью прибавляется/отнимается численное значение эффекта. По внешнему виду аптеки/отравлялки нельзя ничего сказать о силе её воздействия и о знаке её воздействия. После употребления аптеки соответствующая клетка считается просто проходом (повторно съесть аптеку нельзя). В лабиринте могут встречаться другие игроки, которые всегда занимают одну точку пространства (в стенке игрок не может находиться).

В начальный момент игроки расставляются сервером на карту случайным образом. Они должны обязательно оказаться в допустимой точке (то есть не на стенке и не на одной клетке с другим игроком).