

AGH
WIEiT
Systemy operacyjne

Sieciowa gra w statki – Linux

Projekt – dokumentacja

Tymoteusz Machowski

1. Założenia

Celem projektu było napisanie w języku C uproszczonej wersji sieciowej gry w statki. Program miał być przeznaczony na systemy linuxowe oraz miał umożliwiać rozgrywkę wieloosobową.

Wymienione założenia zostały zrealizowane za pomocą różnorodnych mechanizmów niskopoziomowych, takich jak sockety, wątki, muteksy i zmienne warunkowe. Interfejs użytkownika jest uproszczony, zbudowany na ASCII, a gra toczy się w konsoli. Projekt składa się z kilku plików źródłowych: programu serwera (serwer.c), programu klienta (klient.c), pliku konfiguracyjnego (config.h) oraz Makefile.

Przebieg rozgrywki:

- Uruchomienie serwera, o ile nie jest uruchomiony
- Uruchomienie klienta - z podaniem odpowiednich danych: nazwa gracza, protokół komunikacji (unix/inet), ścieżka serwera / adres IP serwera, port serwera
- Łączenie klienta z serwerem
- Menu główne:
 1. Nowa gra
 2. Historia
 3. Wyjście

W zależności od wyboru:

1. Serwer szuka przeciwnika, gdy znajdzie, gra się rozpoczyna. Najpierw gracze ustawiają swoje statki, gdy obaj są gotowi, drugi zaczyna rozgrywkę. Gracze próbują trafić w statek przeciwnika – jeśli trafiają, strzelają dalej, jeśli nie, kolej przeciwnika. Gra toczy się do momentu, gdy jeden z graczy straci wszystkie statki (drugi wygrywa). Po zakończeniu następuje powrót do menu głównego. Serwer natomiast zapisuje wynik w plikach dla obu graczy.
2. Serwer wysyła do klienta jego historię rozgrywek, która jest wyświetlana na ekranie. Następuje powrót do menu głównego.
3. Gra zostaje wyłączona, a gracz wyrejestrowany z serwera.

2. Plik *config.h*

Plik *config.h* jest nagłówkowym plikiem konfiguracyjnym gry. Zawiera pomocne makrodefinicje, określające m.in. maksymalną ilość klientów oraz maksymalną długość nicku klienta, ale także nazwę folderu z historią gier oraz definicję znacznika końca historii. Drugą ważną częścią pliku są definicje struktur danych używanych w projekcie:

- typy wyliczeniowe – określają: dozwolone stany menu, dozwolone stany gry, możliwe stany pola oraz dostępne komunikaty przesyłane i odbierane z serwera.
- struktury – wykorzystano je w celu lepszej organizacji kodu. Istnieją więc: struktura pola planszy, struktura klienta-gracza, struktura zapytania/odpowiedzi do serwera oraz struktura pojedynczej rozgrywki.

Dokładny wygląd struktur i definicji jest dostępny w omawianym pliku *config.h*.

3. Plik *serwer.c*

Plik ten jest kodem źródłowym serwera obsługującego gry.

Serwer jest wielowątkowy. Główny wątek zajmuje się przyjmowaniem połączeń od nowych klientów funkcją *accept* oraz tworzeniem dla każdego osobnych wątków (*pthread_create*).

Następnie każdy z wątków obsługuje komunikację z poszczególnym klientem. Wszystko odbywa się przez sockety. Serwer przyjmuje i wysyła określone w pliku *config.h* typy komunikatów. Dla większości z nich są przygotowane odpowiednie funkcje obsługi, i tak np.: serwer po otrzymaniu komunikatów *PLAYER_READY* od obu graczy przypisanych do danej gry, wywołuje dla nich funkcję *send_start_game*, by powiadomić ich o rozpoczęciu rozgrywki.

Serwer zajmuje się też zapisywaniem historii rozgrywek w katalogu *history*. Tworzy on tam pliki osobno dla każdego gracza, zapisując datę i wynik gry.

Argumenty uruchomienia serwera:

- port – liczba naturalna z zakresu 1024-65535; może być konieczne znalezienie wolnego portu,
- ścieżka do pliku socketu dla komunikacji lokalnej.

Serwer wyłącza się poprzez wysłanie do niego komunikatu SIGTSTP lub SIGINT (odpowiednio: ctrl-z lub ctrl-c).

4. Plik *klient.c*

Plik ten zawiera w sobie główną logikę gry. Jest to również program wielowątkowy – wątek główny inicjuje program i odpowiada za odbieranie i obsługiwanie komunikatów serwera. Drugi wątek odpowiada za samą grę – przyjmuje z klawiatury informacje od gracza i odpowiednio je przetwarza.

Program klienta przechowuje planszę gracza oraz planszę przeciwnika, nie zna jednak rozstawienia jego statków. Są to dwuwymiarowe tablice liczb całkowitych. Plik zawiera wiele funkcji obsługi gry, takich jak np.: sprawdzanie poprawności wprowadzonego statku/strzału, sprawdzanie stanu planszy, sprawdzanie strzału przeciwnika, wypisanie planszy.

Argumenty uruchomienia klienta:

- nazwa gracza – ciąg znaków, najlepiej liter/liczb całkowitych, nie dłuższy niż określa stała *CLIENT_NAME_LENGTH*
- protokół komunikacji – możliwe są 2 wartości: *unix* – do komunikacji lokalnej, gdy serwer i klient działają na jednej maszynie; *inet* – do komunikacji przez sieć
- ścieżka do socketu serwera – tylko w przypadku, gdy poprzedni argument ma wartość *unix*
- adres IP serwera – gdy podano *inet*, adres IP w standardzie IPv4
- port serwera – gdy podano *inet*, liczba całkowita z zakresu 1024-65535. Adres IP i port serwera muszą być uprzednio znane, aby możliwe było połączenie.

Po pomyślnym połączeniu gracz trafia do menu głównego. Dalszy przebieg działania programu jest zgodny ze schematem przedstawionym w punkcie 1. *Założenia*.