Projekt SK

Konrad Baumgart, Jan Borowski

21 grudnia 2011

1 Treść zadania

Projekt 4: Komunikator internetowy typu GG

2 Protokół sieciowy

Każdy klient identyfikowany jest przez swój 2-bajtowy dodatni numer ID, używa swojego hasła do logowania.
Użytkownik ma tylko 2 statusy: zalogowany i niezalogowany.

Logi Serwer wypisuje dane diagnostyczne na standardowe wyjście błędów.

Zapewnienie odczytu pełnych paczek z danymi Za każdym razem zarówno serwer jak i klient poprzedza przesyłane dane informacją o ich długości (w bajtach) zawartą w 4 bajtach. Ogranicza to na przykład maksymalną długość wiadomości w systemie.
Pierwszą częścią danych jest zawsze kod operacji o długości 1 bajta.

Postępowanie w przypadku otrzymania nieprawidłowych danych Gdy serwer otrzyma od klienta dane niezgodne z niniejszą specyfikacją, rozłącza danego klienta.

Port serwera Serwer nasłuchuje pakietów TCP na porcie 4790.

Łączenie Nawiązując połączenie z serwerem klient może albo się zarejestrować, albo zalogować.

Rejestracja KOD: 1
Można się rejestrować jedynie tuż po nawiązaniu połączenia. Klient przesyła swoje hasło zakończone \0. Serwer odpowiada **KOD: 1** wysyłając 2 bajty: numer ID w przypadku sukcesu lub 0 w przypadku błędu, następnie rozłącza się.

Logowanie KOD: 2
Klient wysyła swoje ID (2 bajty) do serwera, po nim zaś swoje hasło zakończone \0. Serwer odpowiada **KOD: 2** wysyłając 1 bajt: 1 w przypadku sukcesu lub 0 w przypadku błędu.

Zakończenie połączenia Zarówno klient jak i serwer mogą zakończyć połączenie po prostu kończąc połączenie TCP.

Poniższe komendy można wykonać tylko będąc zalogowanym.

Sprawdzenie dostępności znajomych KOD: 3
Klient wysyła kilka ID (2-bajtowe, jedno po drugim) do serwera. Serwer odpowiada **KOD: 3** wysyłając wielokrotność 3 bajtów: dla każdego ID o które wystosowano zapytanie 2 bajty zajmuje to ID, zaś w trzecim bajcie jest kod dostępności danego użytkownika.
Można sprawdzić dostępność samego siebie.

Wysłanie wiadomości KOD: 5
Klient wysyła nieujemną ilość osób n (1 bajt) do których wysyła wiadomość. Następnie wysyła $2n$ bajtów - ID odbiorców.
Serwer nie informauje nadawce o powodzeniu operacji wysłania.
Serwer wysyła **KOD: 5** do odbiorców nieujemną ilość osób n (1 bajt) do których wysyłana jest wiadomość. Następnie wysyła 2 bajty - ID nadawcy, oraz $2(n-1)$ bajtów - ID innych odbiorców.
Dzieki temu klient wie, że wiadomość wysłana jest w konferencji.
Jeżeli odbiorca jest niezalogowany, wiadomości do niego są przechowywane na serwerze i są mu przesyłane w momencie gdy się zaloguje.

3 Realizacja

Serwer Stworzyliśmy serwer używając poznanych na zajęciach socketów BSD używając języka C++. By przechowywać dane o użytkownikach, używamy dynamicznych kontenerów z STL. Kod serwera jest krótki, więc pozwoliliśmy sobie go zamknąć w pojedynczym pliku *main.cpp*.

> *Code is the ultimate documentation.*

Klient Klienta stworzyliśmy w języku Java, gdyż byliśmy w stanie potem uruchamić go w środowisku Linux oraz Windows, a także stworzyć wersję dla urządzenia mobilnego z systemem Android.

4 Obsługa programu

4.1 Serwer

Skompilowany przy (użyciu programu make) serwer uruchamia się w konsoli. Po uruchomieniu można obserwować wyjście diagnostyczne na standardowym wyjściu błędów. By łagodnie wyłączyc serwer wystarczy wysłać doń SIGUSR1.

4.2 Klient

Klient uruchamiany jest na maszynie wirtualnej Java. Intuicyjny graficzny interfejs użytkownika pozwala zarejestrować się, zalogować i kożystać z komunikatora.