Sprawozdanie do projektu z Sieci Komputerowych 2

Autorzy: Jakub Raczyński (136787), Julia Tadej (136820)

1. Protokół komunikacyjny

Serwer oraz klient rozpoznają rodzaj przychodzącej wiadomości na podstawie dodawanego na początku jednoznakowego prefiksu. Każda wiadomość jest łańcuchem znaków, składającym się z wspomnianego wcześniej jednoznakowego prefiksu, spacji, długości wiadomości zapisanej w bajtach, kolejnej spacji i właściwej treści wiadomości. Poniżej opisane są typy wiadomości wraz z ich prefiksami:

Dla klienta:

- „P" – wiadomość od serwera oznaczająca wpisanie błędnego hasła przy próbie logowania
- „a" – wiadomość od serwera oznaczająca wpisanie danych użytkownika, który aktualnie jest zalogowany w systemie
- „W" – wiadomość od serwera oznaczająca wpisanie poprawnych danych w polach login oraz hasło
- „l" – wiadomość od serwera zawierająca listę nicków użytkowników będących online
- „i" – wiadomość od serwera informująca, że użytkownik o nicku podanym w wiadomości zalogował się
- „o" – wiadomość od serwera informująca, że użytkownik o nicku podanym w wiadomości wylogował się
- „m" – wiadomość od serwera zawierająca w sobie wiadomość tekstową od innego klienta

Dla serwera:

- „q" – wiadomość od klienta oznaczająca, że klient odłącza się od serwera
- „l" – wiadomość od klienta przekazująca wpisane przez niego login i hasło
- „o" – wiadomość od klienta informująca, że klient się wylogował
- „m" – wiadomość od klienta, którą serwer powinien przekazać do klienta, którego nick jest zamieszczony w wiadomości

Ponadto serwer i klient odczytują wiadomości w pętli, co zapewnia poprawne działanie przy fragmentacji/sklejaniu wiadomości. Pętla odczytująca kończy się, gdy rozmiar wiadomości podany w niej zaraz po prefiksie pokrywa się z rzeczywistym rozmiarem otrzymanej wiadomości.

2. Struktura projektu

Projekt składa się z programu serwera, napisanego w języku C++ oraz z programu klienta, napisanego w języku Java.
Program serwera działa tylko na systemach operacyjnych uniksopodobnych, gdyż wykorzystuje funkcje jądra systemu Linux do obsługi gniazd sieciowych. Do obsługi wielowątkowości korzystaliśmy z biblioteki thread, dostępnej w języku C++. Poza wątkiem głównym, dla każdego klienta będącego online serwer tworzy dedykowany wątek.

Kod programu serwera znajduje się w 3 plikach .cpp oraz dwóch plikach .h. Pliki User.cpp i User.h zawierają w sobie kod klasy User, zawierającej informacje o użytkownikach, którzy kiedykolwiek połączyli się z serwerem. Pliki Server.cpp i Server.h zawierają w sobie kod klasy Server, która odpowiada za wszystkie funkcje i czynności wykonywane przez serwer. Plik main.cpp jest plikiem głównym, w którym tworzona jest instancja klasy Server i uruchamiana jest pętla, w której serwer akceptuje połączenia przychodzące od klientów i tworzy dla nich dedykowane wątki.
Program klienta został napisany w języku Java na platformie Windows. Do tworzenia GUI klienta wykorzystaliśmy framework JavaFX. Program klienta składa się z pięciu klas – 3 z nich obsługują kolejne sceny zawarte w GUI, jedna służy do komunikacji z serwerem, a ostatnia to klasa Main, w której inicjowane są wszystkie elementy GUI oraz dodana jest procedura obsługująca zamknięcie okna. Ponadto układ obiektów na scenach opisany jest w 3 plikach .fxml, a ich wygląd jest opisany w pliku „Talkie.css”. Ponadto w programie klienta tworzony jest wątek pomocniczy służący odczytywaniu przychodzących wiadomości od serwera.

3. Sposób kompilacji i uruchomienia projektu

W folderze serwer znajduje się skrypt o nazwie „comp.sh”. Należy go wykonać. Plik ten wygeneruje plik binarny o nazwie „main”. Po wykonaniu pliku „main” uruchomi się nam działający serwer (możliwe, że użytkownik będzie musiał nadać sobie prawo wykonania utworzonego pliku „main”, najlepiej zrobić to komendą chmod).
Program klienta z kolei najlepiej uruchomić z poziomu IDE IntelliJ. W IDE IntelliJ należy wybrać opcję Open Project i wybrać folder „SignInApp” znajdujący się w folderze „klient”. Nastepnie wystarczy wybrać opcję run – po wybraniu jej powinien uruchomić się program klienta.