

# Sprawozdanie z projektu z przedmiotu Sieci Komputerowe 2

Autorzy:

Witold Andraszyk, 145427

Stanisław Bilewski, 145203

## 1. Opis protokołu komunikacyjnego

Protokół komunikacyjny używany w projekcie to Transmission Control Protocol (TCP), w architekturze klient – serwer. Komunikacja między klientem a serwerem opiera się na wiadomościach rozpoczynających się od znaku „/” (np. wiadomość od serwera do klienta o treści „/USERS add <user>” spowoduje, że do listy użytkowników po stronie klienta dodany zostanie użytkownik <user>, wiadomość „/owner <user>” poinformuje klienta, że właścicielem pokoju jest użytkownik o nazwie <user>). Z założenia wszystkie wiadomości powinny kończyć się znakiem nowej linii. Jeżeli wiadomość po znaku nowej linii się nie kończy to jest to traktowane tak, jakby kilka wiadomości zostało „sklejonych” – klient reaguje wtedy na każdą część oddzieloną znakiem nowej linii jak na osobną wiadomość.

## 2. Opis implementacji

Serwer napisany został w języku C++ z wykorzystaniem BSD sockets. Serwer jest w gruncie rzeczy aplikacją konsolową, która jako argumenty wejściowe przyjmuje adres IP i port, które chcemy przypisać serwerowi oraz nazwę użytkownika będącego właścicielem serwera. Serwer działa współbieżnie, tworząc osobny wątek dla każdego użytkownika.

Klient napisany został w języku C++ z wykorzystaniem BSD sockets oraz wykorzystując (w ramach GUI) framework Qt. Każde okno implementowane jest w osobnej klasie. Klasa „Room” oraz klasy z niej dziedziczące zawierają właściwą implementację klienta. Okna pokoju dzielą się na dwa rodzaje:

- okno użytkownika;
- okno właściciela.

Pokój zakłada się klikając przycisk „Założ” – powoduje to uruchomienie w nowym wątku serwera i wyświetlenie okna właściciela. Następnie użytkownicy mogą dołączać do właśnie założonego pokoju wpisując jego adres IP (razem z portem) oraz klikając przycisk „Dołącz”. Okno pokoju zbudowane jest na dwóch wątkach – jeden (główny) wątek służy do pisania i wysyłania wiadomości, natomiast drugi do obsługi przychodzących wiadomości. W wątku do obsługi wiadomości w nieskończonej pętli wykonuje się polecenie `recv()` mające na celu wychwycić przychodzące wiadomości, a kiedy wiadomość zostanie wychwycona - uruchamiana jest funkcja do obsługi wiadomości.

Pokój właściciela różni się od pokoju użytkownika tym, że posiada opcję wyrzucania użytkowników (właściciel nie ma możliwości wyrzucenia siebie samego). Po wybraniu użytkownika z listy oraz naciśnięciu przycisku „Wyrzuć” do serwera wysłana zostanie wiadomość o treści „/kick <user>” gdzie <user> to nazwa wyrzucanego użytkownika. Serwer następnie (po sprawdzeniu czy wiadomość przychodzi od właściciela) wyrzuci odpowiedniego użytkownika. Ponadto, kiedy pokój właściciela zostanie zamknięty serwer staje się nieczynny i zostaje wyłączony.

### **3. Opis sposobu kompilacji i uruchomienia projektu**

Aby skompilować serwer wystarczy w terminalu systemu operacyjnego Linux wejść do katalogu z kodem serwera i wpisać komendę „g++ -pthread serwer.cpp -o serwer.out”. W katalogu utworzony zostanie plik serwer.out, który jest programem serwera.

Zalecany sposób kompilacji klienta jest otwarcie projektu w IDE QtCreator, a następnie wykonanie operacji build (kombinacją przycisków Ctrl+B). Utworzy to plik „SK\_Project” w katalogu, którego nazwa zaczyna się na „build-SK\_Project” (podaję tylko początek nazwy, jako że pełna nazwa może być różna zależnie od wersji Qt, która była użyta przy kompilacji). Plik „SK\_Project” jest ostatecznym plikiem wykorzystywanym do uruchomienia programu, można go skopiować do innego katalogu, np. katalogu „build” (tak jak jest to zrobione w repozytorium na GitHub). Skompilowany plik serwera o nazwie „serwer.out” musi się znajdować w tym samym katalogu, co plik „SK\_Project”. Następnie można uruchomić program w terminalu za pomocą komendy „./SK\_Project”.