

System wymiany komunikatów typu publish/subscribe.

Błażej Krzyżanek i Maciej Meler

20 stycznia 2020

1 Temat

Tematem naszego programu było stworzenie systemu do wymiany komunikatów typu publish / subscribe. Stworzyliśmy serwer działający na systemach operacyjnych Linux oraz okienkową aplikację klienta, uruchamianą na systemie Windows. Program pozwala na:

1. rejestrację nowego użytkownika,
2. zalogowanie się na istniejące konto,
3. subskrypcję wiadomości na jeden z 10 predefiniowanych tematów,
4. publikowanie nowych treści.

Ponadto serwer informuje użytkownika o wiadomościach rozgłoszonych na subskrybowane tematy, podczas jego nieobecności. Stan serwera jest cyklicznie zapisywany i może być odczytany przy kolejnym uruchomieniu.

2 Protokół komunikacyjny

Aplikacja opiera się na połączenie opierając się o model TCP/IP. Przy pierwszej interakcji klienta z serwerem tworzone jest połączenie przy użyciu gniazd, połączenie to utrzymywane jest aż do zakończenia pracy przez klienta. Klient ma możliwość wysłania 3 typów zapytań:

Login podczas którego wysyłany jest login i hasło użytkownika, umożliwia zalogowanie się na istniejące konto, natomiast jeżeli użytkownik nie istnieje w bazie - tworzy nowego.

Subscribe które zawiera listę tematów, które chce obserwować dany użytkownik, dostępne po zalogowaniu.

Write zawiera temat oraz treść wiadomości, umożliwia rozesłanie wiadomości na podany temat do wszystkich obserwujących użytkowników, dostępne po zalogowaniu.

Typ zapytania rozpoznawany jest po pierwszym znaku wysłanego tekstu. Następnie sprawdzana jest poprawność treści – czy zgadza się struktura przesłanego tekstu, czy podany temat istnieje itp. W przypadku niezgodności lub nierozpoznania zapytania, wysyłany jest odpowiedni komunikat.

Klient nasłuchuje w tle nadchodzących wiadomości i aktualizuje je w głównym widoku aplikacji. Wiadomości wysyłane są przez serwer od razu po wysłaniu przez klienta publikującego lub przy zalogowaniu, jeżeli wiadomości były rozesłane w trakcie nieobecności użytkownika.

3 Implementacja

3.1 Serwer

Serwer został zaimplementowany w języku c++ na systemie Linux. Na samym początku głównej funkcji serwera wczytywane są zapisane wcześniej dane z pliku przy użyciu strumieni. Następnie tworzony jest zbiór deskryptorów gniazd, oraz inicjalizacja listy klientów wartościami pustymi, które reprezentują oznaczają, że nie ma aktualnie połączonych klientów. W kolejnym kroku inicjowane jest główne gniazdo umożliwiające akceptację połączeń z wielu źródeł.

Po inicjalizacji systemu uruchamiana jest nieskończona pętla, oczekuje na aktywność na którymś z gniazd, a następnie:

1. Jeżeli przyszła wiadomość od nowego klienta, akceptuje ją i dodaje do zbioru połączonych klientów.
2. Jeżeli klient się rozłączył, wykonywane są odpowiednie akcje zaznaczające tę sytuację
3. Jeżeli wiadomość przyszła od połączonego klienta, wykonuje odpowiednią akcję opisaną w poprzedniej sekcji.
4. Zmiany zapisywane są do pliku, który może być odczytany przy starcie programu.

3.2 Klient

Klient został napisany w języku Java. Zostało stworzone GUI w JavaFX. Uruchamiana jest za pomocą pliku „Klient.jar”. Aplikacja posiada ekran logowania i rejestracji. Po zarejestrowaniu zostaje wysłany do serwera komunikat z loginem i hasłem, a następnie wybranymi tematami. Po zalogowaniu do serwera jest wysyłany komunikat z loginem i hasłem, a następnie włącza się ekran główny (w przypadku błędnego zalogowania odpowiedni komunikat). W ekranie głównym jest okno, w którym można wysłać wiadomość na dany temat. Jest także lista wiadomości, która odświeża się (za pomocą odpowiedniego wątku klienta) automatycznie co 10 sekund. Można także przeczytać wiadomość za pomocą przycisku Read.

4 Kompilacja i uruchamianie

Serwer można skompilować na systemie Linux używając komendy:

```
g++ server.cpp -o server -std=c++1y
```

Uruchomić:

```
./server
```

Aplikacja klienta spakowana w pliku .jar, uruchomić ją można na systemie Windows z zainstalowaną Javą używając komendy:

```
java -jar klient.jar
```