

System wymiany komunikatów typu publish/subscribe.

Autorzy: Michał Hordecki 145359 i Agnieszka Gapińska 148131

Poznań, 25.01.2023

1. Temat

Tematem naszego projektu było zaimplementowanie serwera i klienta działających w systemie Publish/subscribe. Serwer został skonfigurowany, tak żeby uruchamiać go na systemach operacyjnych Linux, z kolei aplikację klienta uruchamia się w systemie Windows. W ramach projektu dostępne są następujące funkcjonalności:

- Zalogowanie się do systemu
- Zarejestrowanie się jako odbiorcy (konkretnego typu wiadomości)
- Zarejestrowanie nowego typu wiadomości możliwych do publikowania i subskrybowania
- Napisanie i rozesłanie wiadomości konkretnego typu
- Sprawdzenie, jakie są zarejestrowane typy wiadomości
- Odebranie wiadomości zasubskrybowanych typów

2. Protokół komunikacyjny

Aplikacja opiera się o model TCP/IP. Klient łączy się z serwerem poprzez socket. Po ustanowieniu, połączenie utrzymuje się aż klient go nie zerwie wychodząc z aplikacji. Klient komunikuje się z serwerem przy pomocy komunikatów, gdzie pierwszy znak koduje typ wykonywanej akcji, a reszta komunikatu to przesyłane dane. Podczas działania aplikacji klient może wybrać następujące operacje:

- 1 - Rejestracja odbiorcy – zasubskrybowanie jednego z dostępnych typów wiadomości
- 2 - Rejestracja typu wiadomości – zarejestrowanie nowego typu wiadomości
- 3 - Napisz wiadomość – rozesłanie wiadomości danego typu
- 4 - Odbierz wiadomości – odebranie wiadomości subskrybowanych typów
- 5 - Pokaz zarejestrowane typy wiadomości
- 6 - Wyjście

Za każdym razem klient wybiera akcję, jaką chciałby wykonać. W przypadku próby dodania istniejącego typu wiadomości lub wyboru opcji spoza menu, zostanie poinformowany o błędzie. Po wykonaniu akcji wraca z powrotem do menu.

3. Implementacja

a. Serwer

Serwer został zaimplementowany w języku c dla środowiska Linux. Działa współbieżnie w oparciu o model TCP/IP używając selectów. Po uruchomieniu zostaje utworzony zbiór deskryptorów gniazd oraz zbiór klientów (początkowo z wartościami pustymi - brak klientów). Następnie inicjowane jest główne gniazdo umożliwiające połączenie z wieloma klientami.

Po tych krokach następuje pętla, w której serwer czeka na aktywność na którymś z gniazd. Jeśli zostaje wykryta, to zostają wykonane następujące kroki:

- Jeśli przyszła wiadomość od nowego klienta – zostaje on zapisany i dodany do zbioru połączonych
- Jeżeli klient się rozłączył, wykonywane są odpowiednie akcje zaznaczające tę sytuację
- Jeżeli przyszła wiadomość od któregoś z połączonych klientów, to serwer wykonuje przypisaną do tej wiadomości akcję

b. Klient

Klient został napisany w języku c dla systemu operacyjnego Windows. Aplikacja tworzy obiekt typu SOCKET, a następnie przystępuje do próby nawiązania połączenia (w przypadku sukcesu przechodzi do nieskończonej pętli z opisanym wcześniej menu z akcjami do wyboru, a w przypadku niepowodzenia zwraca komunikat o błędzie połączenia z serwerem i się wyłącza).

4. Kompilacja i uruchamianie:

Serwer (system Linux):

```
gcc -Wall serwer_linux.c -o sewer
```

Klient (Windows):

Budowanie i uruchamianie pliku klient_windows.c w IDE Code Blocks