

Mateusz Małowiecki

Opracowanie noty 4.10 (*Example: The
publish-subscribe pattern*) z rozdziału 4
(*Communication*) książki *Distributed Systems*¹

10 kwietnia 2021

¹Van Steen M., Tanenbaum A.S.: *Distributed Systems 3rd edition*.

Wstęp

Wzorzec *publikuj-subskrybuj* (ang. *publish-subscribe pattern*) polega na tym, że klient subskrybuje pewne określone komunikaty publikowane przez serwer. W wyniku tego przekazywane są tylko te komunikaty, które są subskrybowane przez klienta, a komunikaty, których nikt nie subskrybuje, będą utracone. Wzorzec ten omówimy za pomocą prostego przykładu.

Kod serwera

Na rysunku 1 widzimy przykład naiwnego serwera czasu, który publikuje klientowi swój aktualny czas przez gniazdo *PUB*. Czas jest publikowany co 5 sekund do każdego zainteresowanego klienta.

```
1 import zmq, time
2
3 context = zmq.Context()
4 s = context.socket(zmq.PUB)      # create a publisher socket
5 p = "tcp://" + HOST + ":" + PORT # how and where to communicate
6 s.bind(p)                       # bind socket to the address
7 while True:
8     time.sleep(5)                # wait every 5 seconds
9     s.send("TIME " + time.asctime()) # publish the current time
```

Rysunek 1: Przykładowy kod serwera wzorca *publikuj-subskrybuj*

Kod klienta

Kod klienta jest w tym przypadku równie prosty, jak kod serwera, co możemy zaobserwować na rysunku 2. Na początku tworzymy gniazdo *SUB*, które będzie połączone z odpowiednim gniazdem *PUB* serwera. Następnie klient ustawia sobie subskrypcję komunikatów, które mają *TIME* jako swój tag, celem zyskania możliwości odebrania odpowiednich komunikatów. Na koniec klient odbiera pierwsze 5 komunikatów od serwera i drukuje je na standardowym wyjściu. Oczywiście jest to tylko przykład klienta subskrybującego ten serwer. Serwer ten może bowiem mieć wielu różnych subskrybentów i wysyłać komunikaty do każdego z nich.

```
1 import zmq
2
3 context = zmq.Context()
4 s = context.socket(zmq.SUB)      # create a subscriber socket
5 p = "tcp://" + HOST + ":" + PORT # how and where to communicate
6 s.connect(p)                   # connect to the server
7 s.setsockopt(zmq.SUBSCRIBE, "TIME") # subscribe to TIME messages
8
9 for i in range(5): # Five iterations
10     time = s.recv() # receive a message
11     print time
```

Rysunek 2: Przykładowy kod klienta wzorca *publikuj-subskrybuj*