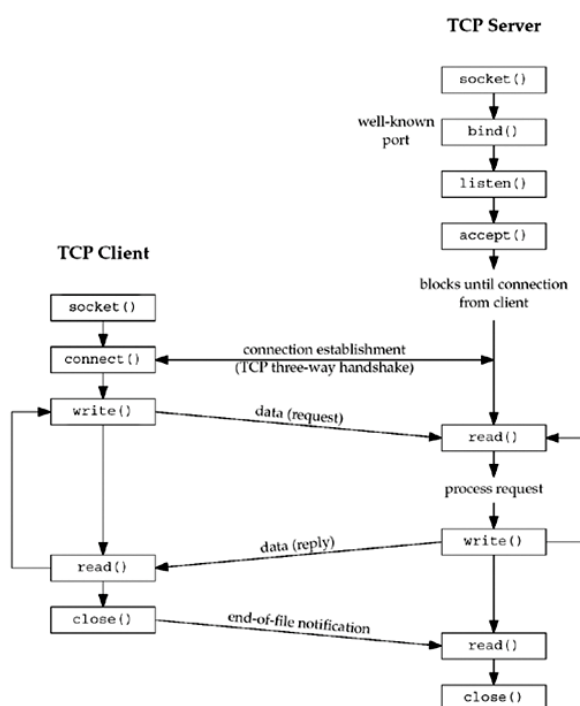


# zaawansowane technologie Javy

## aplikacje typu klient–serwer

Instytut Informatyki  
Uniwersytetu Wrocławskiego

Paweł Rzechonek



TCP jest protokołem działającym w trybie klient–serwer. Serwer oczekuje na nawiązanie połączenia na określonym porcie. Klient inicjalizuje połączenie do serwera.

W przeciwieństwie do UDP, protokół TCP gwarantuje wyższym warstwom komunikacyjnym dostarczenie wszystkich pakietów w całości, z zachowaniem kolejności i bez duplikatów. Zapewnia to wiarygodne połączenie kosztem większego narzutu w postaci nagłówka i większej liczby przesyłanych pakietów. Chociaż protokół definiuje pakiet TCP, z punktu widzenia wyższej warstwy oprogramowania dane płynące połączeniem TCP należy traktować jako ciąg oktetów. W szczególności, jednemu wywołaniu funkcji interfejsu programowania aplikacji `send()`

nie musi odpowiadać wysłanie jednego pakietu. Dane z jednego wywołania mogą zostać podzielone na kilka pakietów lub odwrotnie — dane z kilku wywołań mogą zostać połączone i wysłane jako jeden pakiet. Również funkcje odbierające dane `receive()` w praktyce odbierają nie konkretne pakiety, ale zawartość bufora stosu TCP/IP, wypełnianego sukcesywnie danymi z przychodzących pakietów.

### Zadanie 1

Napisz konsolowy program serwera o nazwie `Date`, który po nawiązaniu połączenia z klientem na protokole TCP na porcie 20251 odpowie komunikatem o bieżącej dacie i godzinie na serwerze. Po udzieleniu odpowiedzi połączenie ma być zamykane a serwer ma oczekiwać na następnego klienta.

Program serwera ma działać w nieskończoność (proces serwera można oczywiście zabić z poziomu administratora).

Działanie serwera przetestuj używając programu `telnet`.

## Zadanie 2

Napisz konsolowy program serwera o nazwie *Echo*, który po nawiązaniu połączenia z klientem na protokole TCP na porcie 20252 będzie odpowiadał takimi samymi komunikatami jakie od niego otrzyma. Dialog taki ma być kontynuowany, aż do momentu wysłania przez klienta linii zawierającej pojedynczą kropkę.

Program serwera ma działać przez pewien określony czas (na przykład przez 1 minutę), a potem ma zwolnić zajmowany port i się wyłączyć.

Działanie serwera przetestuj używając programu *telnet*.

## Zadanie 3

Napisz konsolowy program serwera o nazwie *Totient*, który po nawiązaniu połączenia z klientem na protokole TCP na porcie 20253 będzie mu odpowiadał uporządkowanym ciągiem liczb naturalnych, które są nie większe i względnie pierwsze z zadaną liczbą naturalną (odpowiedź serwera ma oczywisty związek z funkcją  $\Phi$  Eulera). Jeśli przesłane przez klienta dane nie zawierają liczby naturalnej, to serwer ma odpowiedzieć komunikatem o błędzie. Dialog taki ma być kontynuowany, aż do momentu wysłania przez klienta linii zawierającej pojedynczą kropkę.

Program serwera ma działać do momentu, aż operator wpisze polecenie *quit* (stwórz osobny wątek na serwerze, czytający linia po linii polecenia od operatora).

Działanie serwera przetestuj używając programu *telnet*.

## Zadanie 4

Zaprogramuj aplikację sieciową do gry w liczby. Gra ma przebiegać według następującego schematu: serwer po nawiązaniu połączenia z klientem losuje liczbę naturalną z zakresu od 1000 do 9999, a klient stara się ją odgadnąć wysyłając swoje typowania. Serwer odpowiada czy podana liczba jest mniejsza, większa czy równa od propozycji przesłanej przez klienta. Jeśli klient odgadnie wylosowaną przez serwer liczbę przed 13 pytaniem, to zostaje zwycięzcą, w przeciwnym przypadku przegrywa.

Napisz program serwera gry w zgadywanie liczb o nazwie *PickNumber* jako aplikację okienkową w technologii *AWT*, *Swing* albo *JavaFX*. Po nawiązaniu połączenia na protokole TCP na porcie 20254 serwer gra z klientem według opisanych powyżej reguł. Aplikacja powinna mieć możliwość uruchomienia i zatrzymania serwera. Serwer nie kolejkuje połączeń (obsługuje co najwyżej jednego klienta w danym momencie). Przetestuj wstępnie jego działanie za pomocą programu *telnet*.

Napisz także program klienta nazwie *GuessNumber* jako aplikację okienkową w technologii *AWT*, *Swing* albo *JavaFX*. Po nawiązaniu połączenia z serwerem aplikacja klienta ma umożliwiać granie według opisanych powyżej reguł. Klient powinien mieć możliwość poddania się przed rozstrzygnięciem gry.

## Uwaga

Program *telnet* uruchamiamy z dwoma parametrami: nazwą hosta i numerem portu, na przykład:

```
> telnet localhost 20250
```

Program *telnet* działa w taki sposób, że po wysłaniu danych do serwera oczekuje na jego odpowiedź i wypisuje ją na standardowe wyjście. Zaraz po nawiązaniu połączenia *telnet* oczekuje na pierwszą porcję danych (serwer powinien pierwszy napisać coś do klienta).

Program *telnet* jest standardowym programem narzędziowym pod Linuxem. W systemie Windows 11 należy włączyć odpowiednią funkcję, aby *telnet* zadziałał:

- uruchom *Panel sterowania*;
- na stronie głównej *Panelu sterowania* wybierz *Programy*;
- na stronie *Programy* w części *Programy i funkcje* wybierz *Włącz lub wyłącz funkcje systemu Windows*;
- w okienku *Włącz lub wyłącz funkcje systemu Windows* należy zaznaczyć opcję *Telnet Client* i kliknąć przycisk *OK*.