

## Client – Serwer UDP

(do przesyłania plików z kontrolą przepustowości)

### Klient

Klient jest procesem służącym do wysyłania danych.

Parametry uruchomienia klienta:

```
-server <address>
-port <port number>
-file <file path>
```

Po uruchomieniu, klient nawiązuje połączenie z serwerem. Po nawiązaniu połączenia przesyła do serwera informacje o pliku. Potem odbiera od serwera wartość początkowej prędkości przesyłania danych i rozpoczyna przesyłanie pliku określonego jako parametr. Podczas transmisji klient reaguje na odpowiedzi serwera – żądanie zmiany prędkości przesyłania danych, informacje o transmisji.

Podczas przesyłania danych klient wyświetla na bieżąco informacje dotyczące transmisji w postaci:

wysłano B | zostało B | prędkość KB/s | AVG prędkość za 10s KB/s | AVG prędkość

Sent: 18432 B	Remained: 29695 B	Speed: 2.00 KB/s	Speed/10s: <no info>	AVG Speed: 2.00 KB/s
Sent: 20480 B	Remained: 27647 B	Speed: 2.00 KB/s	Speed/10s: 2.00 KB/s	AVG Speed: 2.00 KB/s
Sent: 22528 B	Remained: 25599 B	Speed: 2.00 KB/s	Speed/10s: 2.00 KB/s	AVG Speed: 2.00 KB/s

Po zakończeniu transmisji klient wyświetla podsumowanie transmisji:

- poprawność sumy MD5
- suma MD5
- czas transmisji

Transfer info:

MD5 sum is correct.

MD5 sum:

[ -64, 10, 126, 89, 56, -115, 93, -74, 29, -118, -82, 108, -71, 85, -93, -112 ]

Transfer time: 25.5s

Wykorzystywane klasy: DatagramSocket, DatagramPacket, MessageDigest.

## Serwer

Serwer przyjmuje połączenia klienta i odbiera przesyłany do niego plik.

Parametry uruchomienia serwera:

```
-server <address>  
-port <port number>  
-file <file path>
```

Po uruchomieniu serwer oczekuje na połączenie od klienta, wypisując taka informacje:

```
DatagramSocket address: 10.1.2.206  
DatagramSocket port: 8080  
Waiting for connections...
```

Po nawiązaniu połączenia serwer otrzymuje od klienta nazwę pliku docelowego i jego rozmiar, potem wysyła do klienta wartość początkowej prędkości przesyłania danych i rozpoczyna odbieranie zawartości pliku, zgodnie z protokołem UDP. Podczas tego wyświetla taki komunikat:

```
Receiving file:      zadanie3.pdf  
File size:          46 KB  
Current speed:      2.0 KB/s
```

Podczas pracy serwer może wczytać z klawiatury nową wartość prędkości przesyłania danych. Jeśli taka wartość zostanie wczytana, serwer przekazuje ją do klienta, który przystosowuje się do nowej wartości prędkości. Tego typu zmiany można robić w dowolnej chwili i wielokrotnie w trakcie przesyłania pliku.

```
Enter new speed (KB/s): 5  
Speed changed.  
Current speed:      5.0 KB/s
```

Po zakończeniu transferu serwer przesyła do klienta wyliczoną na podstawie zawartości pliku sumę kontrolną MD5, a po odebraniu potwierdzenia o poprawnym/błędnym transferze, wypisuje odpowiedni komunikat i oczekuje na następne połączenie:

```
Transfer info:  
MD5 sum sent.  
MD5 sum:  
[ -64, 10, 126, 89, 56, -115, 93, -74, 29, -118, -82, 108, -71, 85, -93, -112 ]  
Transfer complete succesfully.  
  
Waiting for connections...
```

Odbierany plik serwer zapisuje na bieżąco na dysku w bieżącym katalogu (jeśli plik o odpowiedniej nazwie istnieje, jego zawartość jest usuwana i nadpisywana).

Wykorzystywane klasy: DatagramSocket, DatagramPacket, MessageDigest.