

11-18.03.2021

- Linuksowe dystrybucje zazwyczaj zawierają program `netcat` (może być też dostępny pod nazwą `nc`) lub jego ulepszoną wersję, `ncat` (ta może być dostępna do zainstalowania jako osobny pakiet albo jako część pakietu `nmap`). Pozwala on m.in. nawiązać połączenie ze wskazanym serwerem, a następnie wysyłać do niego znaki wpisywane z klawiatury; odpowiedzi zwracane przez serwer są drukowane na ekranie. Pozwala też uruchomić się w trybie serwera czekającego na połączenie na wskazanym numerze portu.

Otwórz dwa okna terminalowe, w pierwszym z nich uruchom

```
ncat -v -l 20123
```

a w drugim

```
ncat -v 127.0.0.1 20123
```

(adres `127.0.0.1` to taki magiczny adres IPv4, który zawsze oznacza lokalny komputer). Jeśli wszystko poszło dobrze i netcaty nawiązały połączenie, to linie wpisywane w jednym z okien powinny pojawiać się w drugim. Aby przerwać działanie netcatów użyj Ctrl-C.

Przejrzyj dokumentację netcata, upewnij się co do znaczenia opcji `-v` oraz `-l`. Sprawdź też co robi opcja `-C (--crlf)`, w jakich sytuacjach może być potrzebna?

- Netcat domyślnie korzysta z TCP. Wykonaj powyższe ćwiczenie przy pomocy UDP.
- Programy `nc` oraz `ncat` posiadają opcję `-k` która w połączeniu z `-l` pozwala na obsługę wielokrotnych połączeń. Używając trzech terminali i trzech instancji netcat-a (jedna jako "server" z opcjami `-k -l` i dwie jako "klienci") sprawdź co się dzieje w przypadku dwóch klientów łączących się z tym samym serwerem. Sprawdź co się dzieje osobno dla TCP oraz UDP.
- Napisz prosty serwer zwracający wizytówkę. Powinien tworzyć gniazdko TCP nasłuchujące na porcie o numerze podanym jako `argv[1]` (użyj `socket`, `bind` i `listen`), następnie w pętli czekać na przychodzące połączenia (`accept`), wysyłać ciąg bajtów `"Hello, world!\r\n"` jako swoją wizytówkę, zamykać odebrane połączenie, i wracać na początek pętli. Pętla ma działać w nieskończoność, aby przerwać działanie programu trzeba użyć Ctrl-C.
- Przetestuj netcatem stworzony serwer.

- 
- Napisz prostego klienta, który łączy się (użyj `socket` i `connect`) z usługą wskazaną argumentami podanymi w linii komend (adres IPv4 w `argv[1]`, numer portu TCP w `argv[2]`), drukuje na ekranie wizytówkę zwróconą przez serwer, i kończy pracę. Pamiętaj o zasadzie ograniczonego zaufania i weryfikuj czy odebrane bajty to na pewno drukowalne znaki zanim je prześlesz na stdout.
  - Sprawdź czy program-klient poprawnie współdziała z programem-serwerem.
  - Spróbuj napisać parę klient-serwer komunikującą się przy pomocy protokołu UDP. Pamiętaj o tym, że UDP nie nawiązuje połączeń, więc to klient będzie musiał jako pierwszy wysłać jakiś datagram, a serwer na niego odpowie.
  - (nieobowiązkowe) Przepisz powyższe rozwiązania w innym języku, np. w Javie lub Pythonie. Porównaj obie wersje i oceń, czy nowy kod jest krótszy i/lub czytelniejszy od starego.