

18.03.2021

Na dzisiejszych zajęciach nadal obowiązuje język C/C++. Ostatnie zadanie jest ważne, macie Państwo sześć dni na jego zrobienie i przesłanie poprzez formularz w MS Teams.

- Dokończ pisanie par klient-serwer dla TCP/IPv4 oraz UDP/IPv4 (co razem daje cztery programy). Przetestuj czy działają poprawnie.
- Sprawdź czy klient UDP może wysyłać pusty datagram (o długości 0 bajtów) - jeżeli tak, zaimplementuj takie rozwiązanie informujące serwer o nowym połączeniu
- Rozszerz serwer UDP o wypisywanie na standardowe wyjście adresu IP oraz numeru portu klienta od którego otrzymany został datagram. Przetestuj i sprawdź czy adres i port klienta widziany przez serwer są takie jak się spodziewasz.
- Przeanalizuj niniejszą specyfikację protokołu sumowania liczb. Czy jest ona jednoznaczna, czy też może zostawia pewne rzeczy niedopowiedziane?

Komunikacja pomiędzy klientem a serwerem odbywa się przy pomocy datagramów. Klient wysyła datagram zawierający liczby; serwer odpowiada datagramem zawierającym pojedynczą liczbę (obliczoną sumę) bądź komunikat o błędzie.

Datagramy zawierają znaki ASCII. Datagram przenoszący liczby może zawierać tylko cyfry i spacje; datagram sygnalizujący wystąpienie błędu zawiera pięć znaków składających się na słowo ERROR.

Aby ułatwić ręczne testowanie przy pomocy ncat serwer może również akceptować datagramy mające na końcu dodatkowy znak `\n` albo dwa znaki `\r\n`. Serwer może wtedy, ale nie musi, dodać `\r\n` do zwracanej odpowiedzi.

- Napisz serwer UDP/IP nasłuchujący na porcie nr 2020 i implementujący powyższy protokół. Serwer musi weryfikować odebrane dane i zwracać komunikat o błędzie jeśli są one nieprawidłowe w sensie zgodności ze specyfikacją protokołu.

W kodzie używaj zmiennych roboczych któregoś ze standardowych typów całkowitoliczbowych (`int`, `unsigned long int`, `uint32_t`, itd.). Co za tym idzie, odebrany ciąg cyfr będzie mógł reprezentować liczbę zbyt dużą, aby dało się ją zapisać w zmiennej wybranego typu. Podobnie może się zdarzyć, że podczas dodawania wystąpi przepełnienie (ang. overflow). Serwer ma obowiązek wykrywać takie sytuacje i zwracać błąd. Uwadze Państwa polecam pliki nagłówkowe `limits.h` oraz `stdint.h`, w których znaleźć można m.in. stałą `INT_MAX` czy też `UINT32_MAX`.

To zadanie jest ważne — zaimplementowane rozwiązanie (archiwum zawierające plik źródłowy i np. `makefile`) trzeba przesłać poprzez formularz zadania w MS Teams najpóźniej w czwartek 31 marca.