

# Systemy Operacyjne 2017/18

[Strona główna](#) [Moje kursy](#) [SO2018](#) [Laboratorium 10](#) [Zadania - Zestaw 10](#)

## Zadania - Zestaw 10

**Otwarto:** wtorek, 19 maja 2015, 02:00

Celem zadania jest napisanie prostego systemu pozwalającego na wykonywanie obliczeń na "klastrze obliczeniowym".

Mamy centralny serwer (nasłuchujący jednocześnie na gnieździe sieciowym i gnieździe lokalnym), z poziomu którego zlecane są obliczenia. Klienci po uruchomieniu przesyłają do serwera swoją nazwę, a ten sprawdza czy klient o takiej nazwie już istnieje - jeśli tak, to odsyła klientowi komunikat że nazwa jest już w użyciu; jeśli nie, to zapamiętuje klienta. Zlecenia obliczeń są wpisane bezpośrednio w konsoli serwera i dotyczą czterech podstawowych działań matematycznych - dodawania, odejmowania, mnożenia i dzielenia. Serwer tworzy strukturę opisującą zlecenie (działanie, argument 1, argument 2), a następnie przesyła ją do dowolnie wybranego klienta. Ten oblicza wynik zlecenia i odsyła go do serwera, który wyświetla wynik na standardowym wyjściu. Klient przy wyłączeniu Ctrl+C powinien wyrejestrować się z serwera.

W procesie serwera obsługa wprowadzania zleceń z terminala i obsługa sieci powinny być zrealizowane w dwóch osobnych wątkach. Wątek obsługujący sieć powinien obsługiwać gniazdo sieciowe i gniazdo lokalne jednocześnie, wykorzystując w tym celu funkcje do monitorowania wielu deskryptorów (epoll/poll/select). Dodatkowo, osobny wątek powinien cyklicznie "pingować" zarejestrowanych klientów, aby zweryfikować że wciąż odpowiadają na ządania i jeśli nie - usuwać ich z listy klientów.

Możesz przyjąć, że ilość klientów zarejestrowanych na serwerze jest ograniczona do maksymalnie kilkunastu. Zapewnij możliwość ustalenia dla którego zlecenia wyświetlony został wynik (np. umieść w serwerze globalny licznik, wyświetlany i inkrementowany po dodaniu każdego nowego zlecenia i umieszczany jako dodatkowy identyfikator w komunikatach wysyłanych do klientów i odpowiedziach od nich przychodzących) i który klient go obliczył.

Serwer przyjmuje jako swoje argumenty:

- numer portu TCP/UDP (zależnie od zadania),
- ścieżkę gniazda UNIX.

Klient przyjmuje jako swoje argumenty:

- swoją nazwę (string o z góry ograniczonej długości),
- sposób połączenia z serwerem (sieć lub komunikacja lokalna przez gniazda UNIX),
- adres serwera (adres IPv4 i numer portu lub ścieżkę do gniazda UNIX serwera).

### Zadanie 1

Komunikacja klientów i serwera odbywa się z użyciem protokołu strumieniowego.

**Wskazówka:** na łączy strumieniowym można zasymulować komunikację datagramową korzystając z TLV (Type-Length-Value) - najpierw wysyłana jest stałej wielkości (np. 1 bajt) wartość mówiącą o typie komunikatu/danych, później stałej wielkości (np. 2 bajty) wartość mówiącą o długości przesyłanych danych, aż wreszcie przesyłane są faktyczne dane w ilości określonej chwilę wcześniej; można też np. przechowywać typ danych jako pierwszy bajt przesyłanej wartości - tak jak robią to struktury opisujące adresy gniazd. Użycie tych technik nie jest wymagane w zadaniu.

### Zadanie 2

Komunikacja klientów i serwera odbywa się z użyciem protokołu datagramowego.