

Laborator 1 – Retele de calculatoare

Servere iterative TCP/UDP

1. Functia *connectsock* pe care ati folosit-o in laboratoarele trecute creaza un socket activ. Modificati aceasta functie si scrieti o functie *passivesock* care creeaza un socket pasiv, adica un socket care sa poata fi folosit de un server pentru a trata cereri de conexiuni (*Indicatie*: inspirati-va din slide-ul nr 4 al prezentarii). Scrieti in completare, dupa modelul *connectTCP/connectUDP* de la laboratoarele trecute variantele de creare a unui socket pasiv TCP *passiveTCP*, respectiv UDP *passiveUDP*.

2. Scrieti un server UDP pt serviciul de **time** care sa raspunda la cererile clientului UDP din laboratoarele trecute. Pentru citirea de datagrame UDP folositi functia *recvfrom*, iar pentru scrierea datagramelor UDP folositi functia *sendto*. *Indicatie*: pt a afla timpul, serverul poate apela functia *time*.

De ce nu putem folosi apelurile sistem *read/write*?

3. Scrieti un server pentru serviciul de **daytime** care sa raspunda la cererile clientului TCP din laboratoarele trecute.

4. Scrieti un server pentru serviciul de echo care sa raspunda la cererile clientului TCP din laboratoarele trecute.

Daca ar fi sa scrieti un server concurent, in care fiecare cerere client este servita de un alt proces, creat de server folosind apelul *fork*, cum ati proceda?

5. Modificati serverul **echo** sa foloseasca UDP ca protocol de transport. Simulati conditia de server imposibil de accesat (*unreachable*) pentru a putea testa programul client de la punctul 5 din laboratorul 2.