Laborator 5 – Retele de calculatoare
Servere concurente TCP/UDP

1. Modificati serverul **timed** pt serviciul **time** scris in laboratoarele trecute pentru a raspunde atat la cereri TCP cat si UDP din partea clientilor. *Indicatie*: folositi *select*.

2. Scrieti un server TCP concurent **echod** pentru serviciul de **echo** care sa raspunda la cererile clientului **echo** TCP din laboratoarele trecute. Pentru fiecare cerere client, serverul master creeaza un nou proces (copil, slave) folosind apelul sistem *fork* care va trata separat cererea client, in vreme ce serverul parinte (master) revine la activitatea de acceptare a cererilor clientilor.

Obs: Pentru a opera in mod concurent, serverul parinte nu poate astepta in mod blocant terminarea proceselor (copii) care trateaza cererile client.

3. Folositi un shell script ca in exemplul **daytime** din laboratorul trecut pentru a verifica functionalitatea concurenta a serverului **echod** de mai sus. In acest scop trebuie sa modificati clientul TCP **echo** din laboratorul 2 a.i. sa nu mai citeasca datele de la tastatura ci dintr-un fisier text *echo-input.txt* aflat in directorul curent.

Intrebare: puteti sa cititi datele din fisier si fara sa modificati codul clientului TCP echo?

Nota: Intr-o prima varianta de implementare, pentru a testa functionalitatea serverului, puteti sa considerati ca datele trimise de client au un format standard de tipul “Clientul <PID> vrea sa-si auda ecoul”, unde PID este process id-ul clientului. Acest mesaj poate fi construit dinamic, la run-time, in momentul trimiterii cererii catre server.

Apoi modificati codul clientului sa adauge la mesajul de mai sus textul pe care il citeste din fisierul *echo-input.txt*

4. Modificati serverul **echod** de la punctul 2 pentru a servi si cereri UDP, nu doar TCP.

5. Scrieti un client TCP concurent pentru serviciul **daytime** din laboratorul trecut. Clientul foloseste apelul sistem *fork* pentru a crea mai multe procese client, fiecare dintre ele trimitand o cerere catre serverul **daytimed** cu urmatorul format: “<*n*>”, unde *n* este PID-ul procesului client.

Alternativ, puteti modifica clientul TCP **daytime** din laboratorul trecut in sensul de mai sus si sa folositi un program care foloseste *fork-exec*, parametrul apelului *exec* fiind chiar clientul TCP **daytime** modificat.

6. Modificati serverul **daytimed** din laboratorul trecut a.i. sa poata intelege cererile clientilor TCP **daytime** modificati la punctul 5. Serverul modificat va trimite inapoi clientului PID-ul primit de la client + daytime in formatul “<daytime> for client <PID>”.