

Laborator 9 – Retele de calculatoare

Server Web – partea I

1. Modificati superserverul internet **inetd** a.i. sa trateze cereri TCP ale clientilor de Web pe portul 8080. Serverul propriu-zis de serviciu se va numi **http-echo-inetd**.

Observatie: Superserverul **inetd** redirecteaza socketul de serviciu returnat de apelul *select* in descriptorii de fisier 0, 1 si 2. **Din acest motiv nu se pot folosi functiile obisnuite de lucru cu ecranul si tastatura (gen printf/scanf & co, perror, etc) !** Se poate folosi in schimb apelul *syslog* care va scrie mesajele in */var/log/syslog*

2. Scrieti cod pentru **http-echo-inetd** care citeste o cerere Web a unui client de pe descriptorul de fisier 0 si pregateste un raspuns. Raspunsul trebuie sa inceapa cu un header care in cazul concret al acestui server va consta in urmatoarele string-uri care se vor trimite catre client prin scrierea lor pe descriptorul de fisier 1:

```
"HTTP/1.1 200 OK\n"
```

```
"Content-Type: text/html; charset=utf-8\n"
```

Dupa ce trimite aceste mesaje, **http-echo-inetd** va returna clientului propria lui cerere pe care o va scrie pe descriptorul de fisier 1, dupa care inchide descriptorii 0, 1 si 2 si termina executia.

Testati functionalitatea lui **http-echo-inetd** folosind diverse browsere care acceseaza serviciul folosind URL-ul "http://localhost:8080"

Cititi raspunsul din browser si incercati sa intelegeti ce anume cer clientii unui server Web.

3. Modificati superserverul internet **inetd** a.i. sa trateze cereri TCP ale clientilor de Web (si) pe portul 8081. Serverul propriu-zis de serviciu se va numi **http-inetd**.

4. Scrieti cod pentru **http-inetd** care citeste o cerere Web a unui client de pe descriptorul de fisier 0 si pregateste un raspuns. Raspunsul trebuie sa inceapa cu un header care in cazul concret al acestui server include urmatoarele string-uri care se vor trimite catre client prin scrierea lor pe descriptorul de fisier 1:

```
"HTTP/1.1 200 OK\n"
```

```
"Content-Type: text/html; charset=utf-8\n"
```

Observati ca spre deosebire de versiunea anterioara a serverului, string-ului care desemneaza tipul continutului returnat de server ii lipseste un "\n", pentru ca nu am terminat inca de format headerul care se transmite clientului.

Analizand formatul cererilor client asa cum s-a vazut la punctele anterioare, in comanda GET acestia solicita un anume document Web. Daca folosim URL-ul "http://localhost:8081" acest document este chiar "radacina" (DocumentRoot) ierarhiei de documente html. Implicit, acest fisier se numeste *index.html* (sau *index.php*, de pilda, daca se folosesc documente dinamice).

Creati un fisier *index.html* care contine doar o comanda de redirectare a clientului catre site-ul <https://fmi.unibuc.ro> (cautati pe internet cum se poate crea un asemenea fisier).

Http-inetd va deschide acest fisier cu ajutorul apelului sistem *open* si va determina subsecvent dimensiunea fisierului folosind apelul sistem *fstat* (se poate folosi si *stat*).

Apoi va completa header-ul HTTP de raspuns cu urmatorul string:

```
"Content-Length: <size>\n\n"
```

unde *size* este dimensiunea fisierului *index.html*. In continuare, **http-inetd** va citi continutul fisierului in intregime si il va scrie pe descriptorul de fisier 1 ca raspuns la cererea GET a clientului, dupa care inchide descriptorii 0, 1 si 2.