

Să se implementeze în C/C++ o aplicație client-server (folosind TCP) care simulează protocolul HTTP. Serverul va putea servi simultan mai mulți clienți. Toate comenzile trimise de clienți către server trebuie să fie comenzi HTTP valide, dar nu se cere adăugarea vreunui header. Toate răspunsurile trimise de server către clienți trebuie să fie răspunsuri HTTP valide, dar nu este nevoie de adăugarea vreunui header. Puteți presupune că tot conținutul unei cereri sau al unui răspuns HTTP încapă într-un buffer de 1024 octeți.

Serverul se va porni folosind comanda:

```
./server <PORT>
```

Clienții se vor porni folosind comanda:

```
./client <IP_SERVER> <PORT_SERVER>
```

După conectare, clienții vor putea primi de la tastatură (în orice moment) una din următoarele comenzi:

1. `get <NUME_RESURSA>`
 - a. această comandă va cere de la server resursa dată ca parametru
 - b. resursele pot avea orice nume, iar ierarhia lor este de un singur nivel (veți putea avea resurse de genul „pagina.txt”, dar nu veți putea avea resurse de genul „director/pagina.txt”)
 - c. cererea transmisă de client către server trebuie să fie o cerere HTTP corectă
 - d. exemplu de cerere corectă:
 - i. `GET pagina.txt HTTP/1.1\r\n\r\n`
 - e. exemplu de răspuns de la server dacă resursa „pagina.txt” există:
 - i. `HTTP/1.1 200 OK\r\n\r\nabcdefghijkl`
 - ii. (unde „abcdefghijkl” este conținutul resursei „pagina.txt”)
 - f. exemplu de răspuns de la server dacă resursa „pagina.txt” nu există:
 - i. `HTTP/1.1 404 Not Found\r\n\r\n`
 - g. clientul va afișa pe ecran doar conținutul resursei primite de la server, sau „Pagina căutată nu există” în cazul în care pagina nu există
2. `put <NUME_RESURSA> <CONȚINUT_RESURSA>`
 - a. această comandă va salva pe server resursa dată ca parametru
 - b. resursele pot avea orice nume, iar ierarhia lor este de un singur nivel (veți putea avea resurse de genul „pagina.txt”, dar nu veți putea avea resurse de genul „director/pagina.txt”)
 - c. conținutul resursei care se va salva pe server este un șir de caractere fără spații
 - d. dacă resursa există pe server, ea va fi suprascrisă
 - e. cererea transmisă de client către server trebuie să fie o cerere HTTP corectă
 - f. exemplu de cerere corectă:
 - i. `PUT pagina.txt HTTP/1.1\r\n\r\nabcdefghijkl`
 - ii. (unde „abcdefghijkl” va fi conținutul resursei „pagina.txt” pe server)
 - g. serverul va salva resursa în memorie și va răspunde cu următorul răspuns:
 - i. `HTTP/1.1 200 OK\r\n\r\n`
 - h. clientul va afișa pe ecran mesajul „Pagina a fost salvată” în momentul în care primește confirmarea de la server
3. `delete <NUME_RESURSA>`
 - a. această comandă va cere de la server ștergerea resursei date ca parametru
 - b. resursele pot avea orice nume, iar ierarhia lor este de un singur nivel (veți putea avea resurse de genul „pagina.txt”, dar nu veți putea avea resurse de genul „director/pagina.txt”)
 - c. cererea transmisă de client către server trebuie să fie o cerere HTTP corectă
 - d. exemplu de cerere corectă:
 - i. `DELETE pagina.txt HTTP/1.1\r\n\r\n`
 - e. exemplu de răspuns de la server dacă resursa „pagina.txt” există:
 - i. `HTTP/1.1 200 OK\r\n\r\n`
 - f. exemplu de răspuns de la server dacă resursa „pagina.txt” nu există:
 - i. `HTTP/1.1 404 Not Found\r\n\r\n`

- g. clientul va afișa pe ecran mesajul „Pagina a fost ștearsă”, sau „Pagina dorită nu există” în cazul în care pagina nu există
- 4. quit
 - a. clientul va închide conexiunea cu serverul.

Serverul va putea primi de la tastatură următoarele comenzi:

- 1. status
 - a. se va afișa numărul de clienți conectați și o listă cu numele resurselor existente la server (în orice format)
- 2. quit
 - a. se va închide întreaga aplicație (inclusiv clienții)

Dacă este util, se poate considera că serverul acceptă maxim 10 clienți conectați simultan.