Scrieti in C sau C++ doua aplicatii, server si client, care se conecteaza la un server HTTP si descarca paginile web cerute de utilizator, existand posibilitatea de download recursiv. Serverul implementat va avea rol de centralizare a datelor primite iar clientii vor fi “workeri” care se vor ocupa de conectarea la serverul web si descarcarea efectiva.

1. Aplicatia Server
Aplicatia de tip server are urmatoarele roluri:
Primeste date de intrare de la utilizator (adresa HTTP, etc) prin intermediul liniei de comanda. Aceste date vor fi folosite in momentul in care cel putin un client este conectat
Asteapta ca cel putin 1 client sa se conecteze, pentru a putea incepe descarcarea paginilor. Pe masura ce clientii se vor conecta la acesta, aplicatia server va trebui sa imparta in mod echitabil paginile care trebuie descarcate. Numarul de clienti care se conecteaza este variabil.
Trimite clientilor task-uri de descarcare si asteapta de la acestia paginile dorite. De asemenea, daca modul recursiv este ales de catre utilizator, fiecare client va trimite si o lista de linkuri care au fost gasite in pagina descarcata
Asteapta de la clienti paginile/resursele dorite, precum si o lista de link-uri noi descoperite
Link-urile noi descoperite se vor salva intr-o lista de descarcare si distribuite corespunzator clientilor pentru descarcare
Dupa descarcarea tuturor link-urilor, aplicatia server isi va incheia automat executia. Se considera ca nu mai exista linkuri de descarcat atunci cand lista de descarcare este goala si nu mai exista task-uri aflate in curs de procesare de catre un client

Aplicatia se va apela din linie de comanda astfel:
./server [-r] [-e] [-o <fisier_log>] -p <port>

Optiunile au urmatoarele semnificatii:
-r (recursive): daca aceasta optiune este activata, programul va trebui sa comande descarcarea paginilor in modul recursiv, adica in cazul in care clientii vor raspunde cu o lista de link-uri, acesta va trebui sa le salveze intr-o lista si sa comande descarcarea acestora.

-e (everything): daca aceasta optiune este activata, serverul va trebui sa comande clientilor descarcarea tuturor fisierelor la care se face referire in paginile html, printr-un link de forma .... Se considera ca numele fisierelor vor avea o extensie de 3 sau 4 caractere. Aceasta optiune se poate utiliza in combinatie cu optiunea -r si se vor descarca toate fisierele .zip, .pdf, etc. pentru care exista link-uri in paginile parcurse

-o <fisier_log>: aceasta optiune specifica scrierea mesajelor de status si de eroare in fisiere distincte pe disc, <fisier_log>.stdout respectiv <fisier_log>.stderr. Daca aceasta optiune nu este activata, toate mesajele de status si eroare se vor scrie la iesirea standard (stdout), respectiv iesirea standard de eroare (stderr).
In fisierul .stdout se vor scrie mesaje de status, precum clientii care se conecteaza/deconecteaza, ce pagina/resursa trebuie descarcata de catre un anumit client, etc.
In fisierul .stderr se vor scrie mesajele de eroare. Trebuie sa generati mesaje pentru toate cazurile in care apar erori, si sa specificati in aceste mesajele cauzele erorilor. Daca scrieti programul in C++ este de preferat sa utilizati exceptii.
-p <port>: acest parametru nu este optional si specifica portul pe care aplicatia server va trebui sa asculte pentru conexiuni din partea clientilor. Pana cand cel putin un client nu se va conecta, descarcarea paginilor nu va incepe

Dupa pornirea serverului se pot da urmatoarele comenzi de la tastatura:
status : afiseaza o lista cu clientii conectati in acel moment. Pentru fiecare client se va afisa adresa IP si portul.

download http://<nume_pagina>/<cale_catre_pagina> : se va descarca pagina specificata, conform functionalitatii de mai sus
exit : Serverul anunta clientilor conectati ca urmeaza sa se inchida si pe urma se opreste. Clientii se opresc si ei din rulare.

2. Aplicatia Client
Aplicatia de tip client are urmatoarele roluri:
Primeste in linie de comanda date de la utilizator (adresa server, port server, etc)
Se conecteaza la server, pe adresa si portul specificat
Asteapta de la server task-uri ce reprezinta o pagina/resursa care trebuie descarcata, impreuna cu detalii despre modul de descarcare (recursive/everything)
Trimite fisierele descarcate ca rezultat al procesarii task-ului primit de la server
Asteapta de la server mesajul de inchidere a conexiunii si isi va incheia automat executia. Trebuie sa detecteze si cazul in care serverul se inchide fara a trimite mesajul corespunzator, situatie in care de asemenea isi vor incheia automat executia

Se vor porni mai multe instante de clienti, iar fiecare instanta se va apela din linie de comanda astfel:
./client [-o <fisier_log>] -a <adresa ip server> -p <port>

Optiunile au urmatoarele semnificatii:
-o <fisier_log>: aceasta optiune specifica scrierea mesajelor de status si de eroare in fisiere distincte pe disc, <fisier_log>.stdout respectiv <fisier_log>.stderr. Daca aceasta optiune nu este activata, toate mesajele de status si eroare se vor scrie la iesirea standard (stdout), respectiv iesirea standard de eroare (stderr). Trebuie sa va asigurati ca fiecare client va scrie datele in propriile fisiere de log, folosindu-va de PID-ul fiecarui proces
In fisierul .stdout se vor scrie mesaje de status, precum statusul serverului, daca s-a inceput o secventa de conectare/deconectare de la server, pagina/resursa care trebuie descarcata, etc.
In fisierul .stderr se vor scrie mesajele de eroare. Trebuie sa generati mesaje pentru toate cazurile in care apar erori, si sa specificati in aceste mesajele cauzele erorilor. Daca scrieti programul in C++ este de preferat sa utilizati exceptii.

-a <adresa ip server>: acest parametru nu este optional si specifica adresa IP a statiei pe care exista aplicatia de tip server
-p <port>: acest parametru nu este optional si specifica portul pe care aplicatia server asteapta conexiuni din partea clientilor

3. Alte detalii
Pentru descarcarea propriu-zisa a paginilor, numarul minim de instante de clienti pe care trebuie sa ii folositi este 5, iar numarul maxim este de 10 instante.
In directorul unde se gaseste aplicatia server, se va crea o structura de directoare si fisiere asemanatoare cu cea descarcata, directorul parinte al acesteia avand nume identic cu al serverului HTTP.
De exemplu, daca se cere descarcarea paginii http://site/test/dir1/ceva.html, aplicatia server va trebui sa aiba pe disc urmatoarea cale: site/test/dir1/ceva.html, iar directorul site se va afla in directorul curent.
Se presupune ca link-urile se refera numai la pagini de pe acelasi server, deci nu vor avea forma ..., ci vor fi doar de forma ...
Nivelul maxim de recursivitate pe care trebuie sa il suporte clientii este 5. Se considera ca pagina initiala este pe nivelul 1, paginile referite din ea pe nivelul 2 etc. Veti descarca recursiv numai link-urile catre fisiere html (extensia .html sau .htm) de pe acelasi server, care pot avea una din

urmatoarele forme: ..., ... sau ...
Link-urile catre pagini de pe alte servere (care incep cu "http://") sau catre sectiuni de pagina (care contin caracterul "#") vor fi ignorate. Nu trebuie sa tratati cazul in care pagina HTML contine comentarii, adica atunci cand intalniti un link, nu trebuie sa verificati daca se afla in interiorul unui comentariu.
Pentru a crea directoare din cadrul unui program C/C++, sub Linux, puteti folosi functia system(), care are ca argument o comanda a sistemului de operare (in acest caz va fi comanda mkdir) sau puteti folosi direct apelul de sistem mkdir (man 2 mkdir pentru detalii)
Specificatia protocolului HTTP o gasiti la: http://www.faqs.org/rfcs/rfc1945.html (HTTP 1.0) si http://www.faqs.org/rfcs/rfc2616.html (HTTP 1.1). Puteti utiliza HTTP 1.1 sau HTTP 1.0 (care este mai simplu).

4. Testarea temei
Puteti testa tema conectandu-va la orice server HTTP. O adresa de test, care acopera cerintele temei, este http://www.cs.stir.ac.uk/~kjt/index.html. Pentru a verifica daca tema voastra functioneaza corect, puteti sa incarcati paginile dintr-un browser.
In cadrul testarii, se va folosi un numar variabil de clienti, respectand limitele specificate in sectiunea 3.

5. Continutul arhivei temei
Arhiva temei va contine codul sursa pentru tema impreuna cu un fisier README.txt in care sa descrieti implementarea si un fisier Makefile reprezentand pasii de compilare (trebuie sa existe cel putin regulile de build si clean).