

Ministère de l'éducation, de la culture et de la recherche
de la République de Moldova
Université technique de Moldavie
Faculté d'Ordinateur, Informatique et Microélectronique
Filière francophone "Informatique"



Compte Rendu

Programmation réseau

Travail pratique nr.1

Thème: Socket | HTTP | Multithreading

Effectué par l'étudiant(e) de gr FI-181 :

Bonta Alexandr

Vérifié par le professeur :

Buldumac Oleg

Proiectul pe github: <https://github.com/sasa-bonta/PR>

Scopul:

1. Să se creeze o cerere(request) GET HTTP către web-serverul website-ului me.utm.md utilizând socket.
2. Cu ajutorul expresiilor regulate să se extragă toate imaginile din raspunsul primit din punctul 1.
3. Utilizând Socket, HTTP și firele de execuții să se descarce toate imaginile găsite din punctul 2 într-un folder. Imaginile să se descarce utilizând maximum 4 fire de execuție concomitent.

Pentru nota 9 si 10:

- utilizând 4 fire de execuție nu mai mult de 2 imagini să fie salvate în folder în același timp
- programul elaborat trebuie să funcționeze și pentru utm.md

Atenție:

- Să nu fie utilizate librării HTTP, pentru acest laborator de utilizat sintaxa protocolului HTTP.
- Fiecare cerere(request) GET HTTP să conțină minim 6 antete.
- Pentru punctul 2 alegeți doar JPG, PNG și GIF.
- în rețea totul este la nivel de octeți așa ca înainte de a scrie în socket de asigurat că datele sunt convertite în octeți.
- Pentru utm.md de utilizat portul 443, pentru me.utm.md portul 80.
- Se poate de utilizat semaforul pentru a sincroniza firele de execuție.

Ce este un Socket ?

Socket-ul e un tip de comunicare între un server și un client, în care datele sunt transmise real time(live).

Scopul protocolului http

este un protocol de nivel aplicație scopul lui fiind transferul de date.

Ce este un fir de execuție ?

cea mai mică unitate de procesare ce poate fi programată spre execuție de către sistemul de operare.

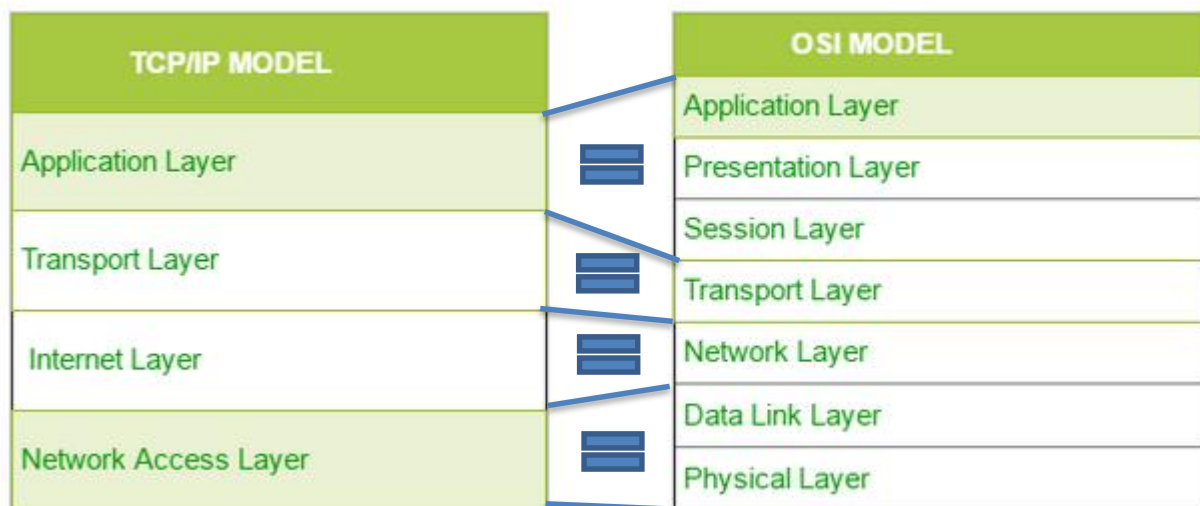
Diferența dintre TCP si UDP

Diferența dintre TCP și UDP - „garanția de livrare“, așa-numitul TCP necesită un răspuns de la clientul căruia este livrat pachetul de date, confirmarea livrării și pentru aceasta are nevoie de o conexiune prestabilă. De asemenea, protocolul TCP este considerat de încredere, în timp ce UDP a primit chiar și numele "protocol de datagrame nesigure". TCP elimină pierderea de date, duplicarea și amestecarea pachetelor, întârzieri. UDP permite toate acestea, și nu are nevoie de o conexiune pentru lucru. Procesele la care datele sunt transmise prin UDP trebuie să gestioneze primite, chiar și cu pierderile. TCP controlează congestia conexiunii, UDP nu controlează decât integritatea datagramei recepționate.

Ordinea în care octeții sunt secvențializați în rețea

32 biti 4 cate 1 octet.

Cele patru nivele ale stivei TCP/IP și șapte nivele ale modelului OSI



Din ce cauză nu puteți transmite cereri HTTP website-ului utm.md utilizând portul 80 ?

Website-ul utm.md, lucrează pe browserul HTTPS, requesturile pot fi trimise pe website-ul dat, însă cu portul 443.

Cum se creează o conexiune Socket ?

`socket(proto, type, family)`

Cum se transmit date într-un Socket ?

`send(socket, data, length, flags)`

Cum se citesc datele dintr-un Socket ?

`recv(socket, buffer, length, flags)`

Care sunt metodele HTTP?

1. **GET**: este cea mai folosită metodă, fiind utilizată atunci când serverului i se cere o resursă.
2. **HEAD**: se comportă exact ca metoda GET, dar serverul returnează doar antetul resursei, ceea ce permite clientului să inspecteze antetul resursei, fără a fi nevoit să obțină și corpul resursei.
3. **PUT**: metoda este folosită pentru a depune documente pe server, fiind inversul metodei GET.
4. **POST**: a fost proiectată pentru a trimite date de intrare către server.
5. **DELETE**: este opusul metodei PUT.
6. **TRACE**: este o metodă folosită de obicei pentru diagnosticare, putând da mai multe informații despre traseul urmat de legătura HTTP, fiecare server proxy adăugându-și semnătura în antetul Via.
7. **OPTIONS**: este folosită pentru identificarea capacităților serverului Web, înainte de a face o cerere.
8. **CONNECT**: este o metodă folosită în general de serverele intermediare.

Codurile de stare HTTP (200, 301, 302, 401, 404, 405, 500)

<https://http.cat/>

200 – OK – cod de răspuns, care indică faptul că solicitarea a fost finalizată cu succes.

301 – Moved Permanently – cod de redirectionare, care indica faptul ca resursa solicitata a fost mutata permanent, la adresa URL specificata in antetul Locatie.

302 – Found – cod de redirectionare, inseamna ca resursa solicitata a fost mutata temporar la adresa specificata in antetul de Locatie. Dupa ce a primit un astfel de raspuns, browserul este redirectionat catre o noua pagina, iar motorul de cautare nu isi actualizeaza legaturile catre resursa nu se schimba si nu est trimisa la o noua adresa URL

401 – Unauthorized – client error status, indica faptul ca soliciatrea nu a fost aplicata, deoarece nu esti autorizat pe resursa data si lipsescacreditari valide pentru resursa data. Este similara cu 403, doar ca in cazul dat autorizarea este posibila.

404 – Not Found – client error status, indica faptul ca serverul nu poate gasi resursa ceruta. Acest stats code, cel mai probabil cel mai cunoscut in retea.

405 – Method Not Allowed – status code HTTP, indica faptul ca metoda solicitarii este cunoscuta de server, DAR a fost dezactivata si nu poatefi utilizata. Doua metode GET si HEAD nu trebuie niciodata dezactivate si nu ar trebuie sa returneze acest cod de eroare. Serverul trebuie sa genereze un camp cu antet Allow in raspunsul cu codul 405, care contine lista metodelor de resurse disponibile.

500 – Internal Server Error – server error response code, indica faptul ca serverul a intalnit o conditie neasteptata care l-a impiedicat sa execute cererea.

Ce este HTTP securizat ?

S-HTTP criptează doar datele paginii difuzate și a transmis date precum câmpurile POST, lăsând inițiativa protocolului neschimbată. Din această cauză, S-HTTP ar putea fi utilizat concomitent cu HTTP (nesecurizat) pe același port, întrucât antetul necriptat ar determina dacă restul transmisiei este criptat.

Diferența dintre HTTP și HTTPS

Hypertext Transfer Protocol (HTTP) : In mod normal este folosit protocolul HTTP pentru a trimite cereri catre server. in acest caz adresa web va incepe cu 'http://' si va folosi portul 80 pentru a comunica cu serverul. Acest protocol nu foloseste nici-un fel de criptare a datelor, deci nu este recomandat sa folosim acest protocol pentru trimiterea informatiilor importante (date personale, date card, date cont bancar, etc).

Hypertext Transfer Protocol Secure (HTTPS) :HTTPS este de fapt o combinatie intre doua protocoale diferite, Hypertext Transfer Protocol (HTTPS) si SSL/TLS. Link-urile incep cu 'https://'. Acest protocol ne ofera o metoda sigura de a transmite datele importante.Acest protocol foloseste portul 443 pentru a comunica cu serverul. HTTPS are un algoritm de criptare ce asigura siguranta datelor.

Diferența dintre Socket și WebSocket

Chiar dacă realizează (în general) lucruri similare, ele sunt cu adevărat diferite. WebSocket-urile rulează de obicei din browserele care se conectează la Application Server printr-un protocol similar cu HTTP care rulează pe TCP / IP. Deci, acestea sunt destinate în principal aplicațiilor web care necesită o conexiune permanentă la serverul său. Pe de altă parte, prizele simple sunt mai puternice și mai generice. Acestea rulează pe TCP / IP, dar nu sunt limitate la browsere sau protocol HTTP. Acestea ar putea fi utilizate pentru a implementa orice fel de comunicare.

Diferența dintre un proces și un fir de execuție

Un fir de executie poate exista doar intr-un proces

Proces usor, Context de executie

Proces nou: cod + date

Fir nou : cod

un program poate utiliza mai multe fire de execuție pe când același program poate utiliza un singur proces.

Ce este un Deadlock și când el apare ?

Deadlock este o situație în care un set de procese sunt blocate, deoarece fiecare proces deține o resursă și așteaptă o altă resursă dobândită de un alt proces.

Deadlock poate apărea dacă următoarele patru condiții se mențin simultan:

Mutual Exclusion: Una sau mai multe resurse nu pot fi folosite (un singur proces poate fi utilizat simultan)

Hold and Wait: un proces reține cel puțin o resursă și așteaptă resurse.

No Preemption: o resursă nu poate fi preluată dintr-un proces decât dacă procesul eliberează resursa.

Circular Wait: Un set de procese se așteaptă reciproc în formă ciclică.

Pentru ce este nevoie de metoda join() în programarea multithreading?

Metoda join() permite unui thread să se poziționeze la sfârșitul altui thread ("join onto the end").

Proiectul pe github: <https://github.com/sasa-bonta/PR>