**Ce este un Socket ?**

Un socket reprezinta un punct de conexiune intro retea TCPIP. Cand doua programe aflate pe doua calculatoare in retea doresc sa comunice, fiecare dintre ele utilizeaza un socket. Unul dintre programe (serverul) va deschide un socket si va astepta conexiuni, iar celalalt program (clientul), se va conecta la server si astfel schimbul de informatii poate incepe. Pentru a stabili o conexiune, clientul va trebui sa cunoasca adresa destinatiei ( a pc-ului pe care este deschis socket-ul) si portul pe care socketul este deschis.

**Scopul protocolului HTTP**

Scopul la HTTP este de a procesa datele, colectare sau transfer, este un protocol de nivel aplicație; La fel este metoda cea mai des utilizată pentru accesarea informațiilor în Internet.

**Ce este un fir de execuție ?**

Un fir de executie este unitatea de executie a unui proces. Fiecarui fir de executie i se asociaza o secventa de instructiuni, un set de registri CPU si o stiva. Procesul nu executa instructiuni, este un spatiu de adresare comun pentru unul sau mai multe fire de executie. Firele de executie sunt cele care executa instructiunile.

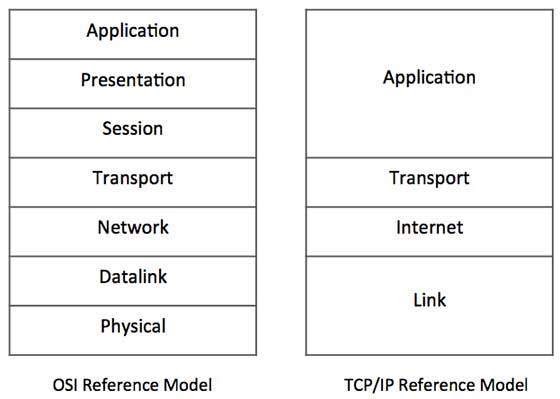
**Diferența dintre TCP si UDP**

TCP este bazat mai mult pe procesarea, transmiterea datelor corect, facand verificarea dintre datele trimise si receptionate pe cand UDP este bazat pe viteza transmiterii datelor, fara a face verificare.

**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 utilizînd portul 80 ?**

Website-ul utm.md este securizat

**Cum se creează o conexiune Socket ?**

var socket(protofamily, type, protocol)

**Cum se transmit date intr-un Socket ?**

send(socket, data, length, flags)

**Cum se citesc datele dintr-un Socket ?**

recv(socket, buffer, length, flags)

**Care sunt metodele HTTP?**

* GET: este cea mai folosită metodă, fiind utilizată atunci când serverului i se cere o resursă.
* 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.
* PUT: metoda este folosită pentru a depune documente pe [server](https://ro.wikipedia.org/wiki/Server), fiind inversul metodei GET.
* POST: a fost proiectată pentru a trimite date de intrare către server.
* DELETE: este opusul metodei PUT.
* 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.
* OPTIONS: este folosită pentru identificarea capacităților serverului Web, înainte de a face o cerere.
* CONNECT: este o metodă folosită în general de serverele intermediare.

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

200 - ok:  
 Această cerere a fost executată cu succes. Informația a revenit cu un răspuns pozitiv, indiferent de modul în care s-a făcut cererea.

301 - modificat/mutat permanent:  
 Cererii i-a fost atribuite o sursă nouă și permanentă URI iar cererile următoare ar trebui să folosească una din sursele returnate URI. Dacă acest mesaj/cod este primit ca răspuns al unei cereri tip „Primit/recepționat” sau „Direcționat/condus”, browser-ul nu trebuie să redirecționeze automat cererea, doar dacă nu poate fi confirmată de către utilizator.

302 - găsit:

Sursa cererii este temporar pe un alt URI. În cazul în care redirectarea ar putea fi schimbată ocazional, utilizatorul ar trebui să folosească în continuare cererea URI (Request-URI) în cazul unor cereri ulterioare.

Dacă mesajul/statusul 302 este recepționat ca răspuns al unei cereri alta decât „Primit/recepționat” sau „Direcționat/condus”, browser-ul nu trebuie să redirecteze automat cererea dacă aceasta nu poate fi confirmată de cărte utilizator.

401 - neautorizat:  
Cererea necesită autentificarea/înregistrarea utilizatorului. Răspunsul trebuie să includă WWW - câmp de autentificare conținând o somație aplicabilă utilizatorului. Utilizatorul poate repeta cererea. Dacă cererea deja includea câmpul de autorizare, atunci raspunsul 401 indică faptul că autorizarea a fost refuzată. Dacă răspunsul 401 conține aceeași somație ca răspuns principal iar browser-ul a executat autentificarea cel puțin o dată, atunci utilizatorul ar trebui să i se prezinte intrarea dată în răspuns, din moment ce aceasta include informații relevante.

404 - negăsit:  
Serverul nu a găsit nimic care să corespundă cererii URI. Nu se dau indicații despre condiția temporară sau permanentă.

405 - metodă nepermisă:  
Metoda specificată în linia de cerere (Request-Line) nu este permisă de către sursa identificată după URI-ul cerut.

500 – Internal Server Error

 indică faptul că serverul a întâlnit o condiție neașteptată care l-a împiedicat să completeze solicitarea.

Această eroare este o generalizare a răspunsului la captarea tuturor excepțiilor („catch-all”). Adesea, administratorii serverului înregistrează (log) mesaje de eroare, cum ar fi codul de stare 500, cu informații mai detaliate despre cerere, pentru a preveni repetarea unei erori în viitor.

**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, da, sunt cu adevărat diferite. WebSockets rulează de obicei din browserele care se conectează la Application Server printr-un protocol similar cu HTTPthat care rulează TCP / IP. Deci sunt în principal pentru aplicații web care necesită o conexiune permanentă la serverul său. Pe de altă parte, prize simple sunt mai puternice și mai generice. Ei rulează TCP / IP, dar nu sunt restricționate la browsere sau HTTPprotocol. Ele ar putea fi folosite 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 executie pe cand acelasi program poate utiliza un singur proces.

**Ce este un Deadlock și când el apare ?**

Este foarte posibil intr-un mediu multi-utilizator, ca utilizatorii isi vor bloca unii altora resursele. Este de asemenea posibil ca doi utilizatori sa sfarseasca prin a-si bloca unul altuia resurse diferite. Aceasta situatie se numeste DEADLOCK deoarece fiecare utilizator asteapta sesursele detinute de celalalt utilizator.

**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”)**.**