

Laboratorul 4. Programarea în rețea

Aplicație Client-Server TCP

Să se creeze o aplicație Client-Server TCP utilizând Sockets API

Pentru nota 8, 9 și 10:

- Să proiectați și elaborați propriul protocol

Atenție:

- Fiecare client trebuie procesat de către server într-un fir de execuție aparte
- Nu se admite aplicații simple de genul Echo Client-Server
- Aplicația poate fi GUI sau consolă
- Pentru cei ce doresc 8, 9 și 10, ca exemplu luați protocoalele HTTP, SMTP unde clientul și serverul discută printr-un set de reguli bine definite. De exemplu, pentru o aplicație de tip chat pentru ca clientul să fie identificat de către server acesta trebuie să transmită un mesaj de tipul: HELLO-REQUEST după care serverul va răspunde cu HELLO-APPROVE. Mai detaliat accesați ultimele 3 referințe de la sfârșitul acestui fișier.

Întrebări la apărarea laboratorului:

- Ce este un protocol orientat pe conexiune ?
- Ce tipuri de aplicații beneficiază în general de utilizarea protocolului TCP ?
- Cum TCP garantează că datele vor fi transmise cu succes ?
- Diferența dintre blocking și non-blocking sockets
- Diferența dintre blocking multithreaded și non-blocking single thread socket
- Cum are loc procesul TCP Three Way Handshake ?
- Numiți cele 4 apeluri de sistem necesare pentru a crea un server TCP
- Care este rolul metodei **bind()** ?
- Care este rolul metodei **accept()** ?
- Ce se întâmplă când apeleți mai întâi **connect()** apoi **bind()** ?
- Ați avea vreodată nevoie să implementați un timeout într-un client sau server care utilizează TCP ?
- Într-o conexiune TCP, clientul sau serverul trimite mai întâi datele ?
- Care este adresa de loopback IPv4 și care este rolul ei ?
- De unde știe un sistem de operare ce aplicație este responsabilă pentru un pachet primit din rețea ?
- Datele primite prin **recv()** au întotdeauna aceeași dimensiune cu datele trimise cu **send()** ?
- Este acceptabil să închei execuția programului dacă este detectată o eroare de rețea ?
- Puteți îmbunătăți performanța aplicației prin dezactivarea algoritmului Nagle ?
- Ce instrumente listează socket-urile TCP deschise în sistemele de operare Windows și Linux ?
- Tehnici de sincronizare a firelor de execuții

Link-uri utile:

- <https://www.thegeekstuff.com/2011/11/tcp-ip-fundamentals/>
- <https://stackoverflow.com/questions/5330277/what-are-examples-of-tcp-and-udp-in-real-life>
- <https://codeburst.io/understanding-tcp-internals-step-by-step-for-software-engineers-system-designers-part-1-df0c10b86449>
- <https://stackoverflow.com/questions/6187456/tcp-vs-udp-on-video-stream?rq=1>
- <https://stackoverflow.com/questions/10654286/why-should-i-use-non-blocking-or-blocking-sockets>
- <https://luminousmen.com/post/asynchronous-programming-blocking-and-non-blocking>
- https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.hala001/orgblockasyn.htm
- <https://www.cs.dartmouth.edu/~campbell/cs50/socketprogramming.html>
- https://www.inetdaemon.com/tutorials/internet/tcp/3-way_handshake.shtml
- <https://hackernoon.com/tcp-three-way-handshake-4161eb8aba32>
- <https://github.com/angrave/SystemProgramming/wiki/Networking%2C-Part-4%3A-Building-a-simple-TCP-Server>
- <https://www.lifewire.com/network-computer-special-ip-address-818385>
- <https://www.extrahop.com/company/blog/2016/tcp-nodelay-nagle-quickack-best-practices/>
- <http://www.cs.unc.edu/~dewan/734/current/lectures/2-Synchronization.pdf>
- <http://shazsterblog.blogspot.com/2011/12/comparison-of-countdownlatch.html>
- <https://habr.com/ru/post/277669/>
- https://link.springer.com/content/pdf/10.1007%2F3-540-45812-3_21.pdf
- https://www.egr.msu.edu/classes/ece480/capstone/fall12/group02/documents/Ryan-Lattrel_App-Note.pdf
- <https://tools.ietf.org/html/rfc3117>