

Labb 1

Ett enkelt chatt-system

Förberedelser

Se till att du följer de allmänna anvisningarna för varje labb innan du börjar med själva laborerandet.

Översikt

I den här laborationen ska du utveckla ett chat-system, följande krav måste uppfyllas:

1. Systemet ska skrivas i programspråket Java och klasserna `java.net.Socket` och `java.net.ServerSocket` måste användas i programmet.
2. Systemet ska kunna driftsättas på två maskiner med olika IP-adresser, Servern och minst en klient körs på en av de maskinerna och minst en klient körs på den andra maskinen.
3. Systemet får inte vara blockerande varken på anslutningsnivå eller på meddelandesändningsnivå. Alltså flera klienter ska samtidigt kunna ansluta sig till servern utan att någon klient blir blockerad på något sätt. Dessutom utgående och inkommande meddelanden får inte heller blockeras.
4. Servern måste kunna samtidigt hantera flera klienter utan att riskera meddelandesändningsmekanismen.
5. Ordning mellan inkommande meddelanden hos klienterna ska vara exakt samma ordning som meddelanden erhållits av servern.
6. Servern måste bifoga avsändarens nicknamn i meddelandet som skickas till mottagar-klienter. Med andra ord när ett meddelande tas emot av servern ska den INTE innehålla avsändarens nicknamn men innan servern skickar meddelandet vidare till anslutna klienter så bifogar servern avsändaren nicknamn i meddelandet. Och det är **viktigt att detta utförs av servern**, detta för att man ska vara säker på att meddelanden är autentiska, d.v.s ingen klient ska kunna låtsat skicka meddelanden med någon annans namn.
7. Om servern tar emot ett meddelande som inte har information om någon mottagande klient, då skickar servern meddelandet vidare till alla klienter (även avsändaren). Men om meddelandet innehåller information om någon mottagande klient då skickas meddelandet endast till mottagar-klienten.

Tips:

- På servern behöver du ha en lista med alla socket-instanser.
- Använd SSH för fjärranslutning till en annan maskin för att sedan kunna starta en klient. Exempelvis genom SSH:a till `myUserName@u-shell.csc.kth.se` och starta en klient därifrån.

Referenser:

- För mer information om hur man skapar och hanterar trådar:
<http://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>
- För mer grundläggande information om hur grundläggande IO fungerar i java:
<http://docs.oracle.com/javase/tutorial/essential/io/index.html>

Frågor:

1. Om du har **N** anslutna klienter, hur många instanser av klassen `Thread` behövs i:
 - a. Server?
 - b. Klient?
2. Vad är `Runnable` i Java?
3. Vad är nyckelordet `synchronized` i java?
4. Beskriv de fyra skikten i TCP / IP-protokollstacken.
5. Vad betyder flaggorna, ACK, SYN och SEQ och vilket protokoll hör de till?
6. Vad är skillnaden mellan TCP och UDP?