

TDDD78 projekt: IRC chatt-klient

1 Introduktion

Internet Relay Chat (IRC) är ett protokoll för chattmeddelanden. Protokollet använder sig av en klient-server-arkitektur. Det vill säga det finns en server som flera klienter kommunicerar med istället för att kommunicera mellan varandra. Protokollet tillåter att man skriver till kanaler, öppna meddelanden för alla på kanalen, skriver direktmeddelanden till andra användare och mycket mer.

Tänk på att det här projektet kräver att ni arbetar med nätverksprogrammering. Det är inget som ingår i kursen och ni kommer därför vara tvungna att läsa på lite extra.

2 Bakgrund

Se Wikipedias artikel: https://en.wikipedia.org/wiki/Internet_Relay_Chat.

Samt IRC protokollet för klienter: <https://tools.ietf.org/html/rfc2812>

3 Milstolpar

IRC protokollet för klienter täcker in väldigt mycket så därför finns det stora möjligheter att utöka projektet med mer. Figur 1 täcker in lite av den grundläggande funktionaliteten i några milstolpar.

3.1 Milstolpe beskrivningar

Följande är beskrivningar av de milstolpar som är med i projektet (se figur 1):

- **Bekanta er med protokollet:** Innan ni börjar att designa och implementera er chatt-klient bör ni sätta er in i grunderna i IRC protokollet. Försumma inte den här punkten då protokollets utformning kan påverka er design avsevärt.
- **Anslut till server:** I den här milstolpen ska ni ansluta till en server via er Java kod. Än så länge räcker det gott och väl med att ni ansluter till en hårdkodad server och ni behöver inte skapa ett fönster ännu.
- **Gå med i kanaler:** Efter att ni har anslutit till en server är nästa steg att gå med i en kanal på servern. Tills vidare kan ni även låta den här biten vara hårdkodad.
- **Val av server:** Det här är en av de första milstolparna där ni låter användarna ange vart de vill ansluta sig. Tänk på att användarna kan vilja byta vilken server de är ansluten till.

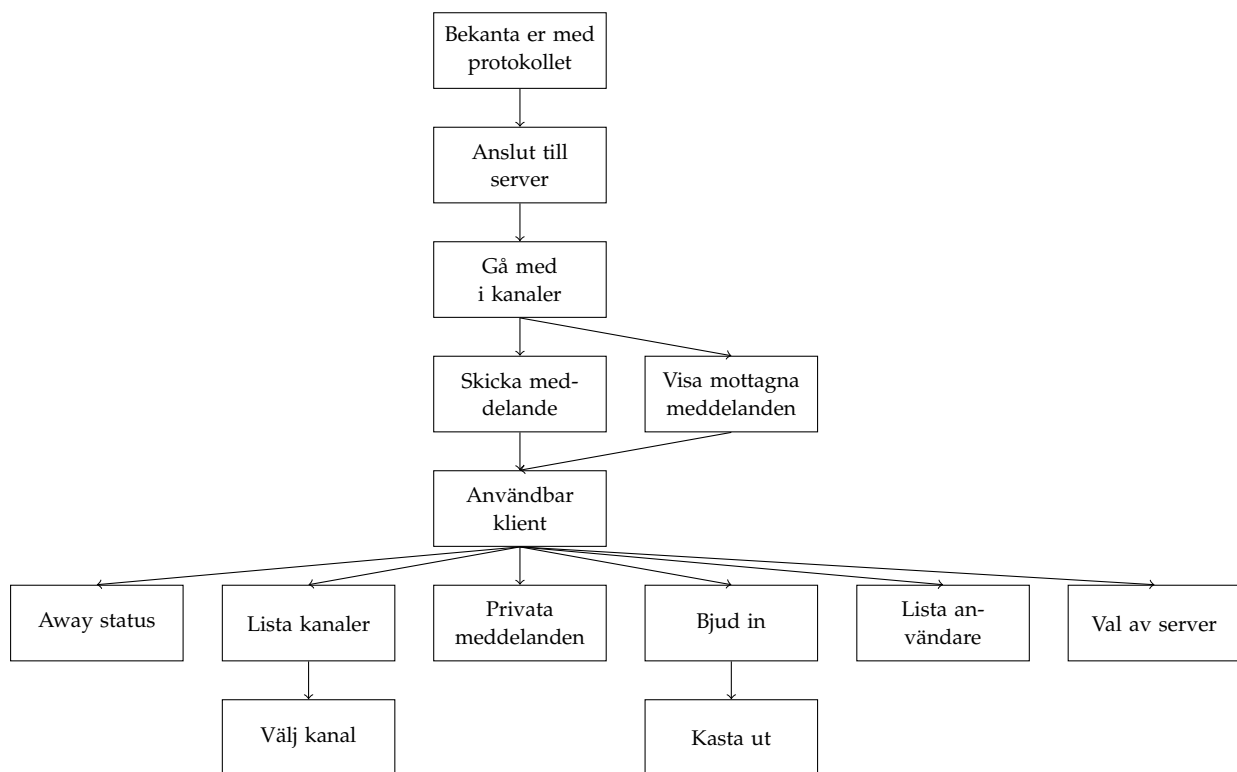


Fig. 1: De första milstolparna.

- **Skicka meddelanden:** När man har anslutit till en server och en kanal är det bra att kunna skicka meddelanden till de andra som är anslutna till kanalen och servern. Det är poängen med den här milstolpen.
- **Visa mottagna meddelanden:** Lika viktigt (eller kanske viktigare) som att skicka meddelanden är att kunna ta emot meddelanden. Därför är den här milstolpen som går ut på att visa meddelandena som skickas till den kanalen användaren är med i och privata meddelandena essentiell för klienten.
- **Användbar klient:** Nu bör ni ha en någorlunda användbar klient. Stanna upp på den här milstolpen och testa ert program för att se till att det fungerar korrekt. Fixa eventuella problem innan ni fortsätter.
- **Away status:** IRC har stöd för att man ska kunna sätta sin status till "Away" för att visa att man inte läser meddelanden av någon anledning (som kan anges). Det är något som kan vara av intresse i en chatt-klient. Den här milstolpen syftar till att låta användaren ställa in sin status till Away samt vilket meddelande som ska visas.
- **Lista kanaler:** Ett första steg för att låta användaren välja vilken kanal hen ska gå med i på en server är att lista vilka som finns tillgängliga. Syftet med den här milstolpen är att lista alla kanaler som finns på den nuvarande servern. Tänk på att vilka kanaler som finns kan ändras med tiden varpå listan måste uppdateras.

- **Privata meddelande:** En vanlig funktion i chatt-klienter i allmänhet är att kunna skriva privata meddelande direkt till andra användare. Det går att göra i IRC också så därför ska klienten i den här målstolpen utökas med den funktionaliteten.
- **Bjud in:** IRC protokollet har möjligheten att bjuda in användare till kanaler (och därmed skapa nya ifall kanalen inte redan finns). Syftet med den här milstolpen är att ge användaren möjlighet att bjuda in en eller fler användare till en kanal.
- **Kasta ut:** Steget efter att man kan bjuda in personer till en kanal är naturligtvis att kunna kasta ut dem igen. Det här bör även kunna göras på ett smidigt sätt vilket även är målet med den här milstolpen.
- **Lista användare:** Många kommandon som finns tillgängliga blir betydligt kraftfullare om man har möjligheten att lista alla användare. Det finns två alternativ för att lista användare. Den ena är att lista alla användare som finns på kanalen och den andra är att lista alla användare på servern. Båda kan vara önskvärda så den här milstolpen inbegriper båda.

4 Designbeslut att tänka på

Det finns ett designbeslut som kan vara värt att tänka på: vad för typ av gränssnitt ska chatt-klienten ha. Klienten kan antingen ha ett gränssnitt som är grafiskt eller ett som är baserat på terminalen (kallat command line interface eller CLI). Ni bör även vara medvetna om att ett CLI troligen innebär mer arbete, speciellt om gränssnittet ska bli bra, eftersom det kräver att man måste hantera mer ostrukturerade indata som användaren skriver in. Vidare måste man lyckas presentera alla data på bra sätt. Ett (icke-testat i den här kursen) alternativ för att göra ett terminalbaserat interface är paketet lanterns.