

Dokumentasjon Oppgave 7

Denne oppgaven var ganske grei ettersom jeg hadde gjort ganske mye av oppgaven fra før av.

Så blir litt rart å si fremgangsmåten på noe av det her ettersom noe av det jeg leverer nå er gjort fra før.

Men det jeg har gjort var å lage en mappe for server og en for client, og gjøre klart makefiles osv..

Server filen:

I main() så looper jeg først igjennom hele argv[] som man sender med når man skal kjøre koden, og lagrer port og id om man har skrevet -port og -id etterfulgt av da portnr/id-navn.

Så lager jeg en socket, og så setter jeg client settings, så bind() man socket med en adress, etterfulgt av listen() for å lytte på socketen etter connections, så accept() man til slutt, som da venter til en client har knyttet seg. Etter det så kjører jeg funksjonen som skal "snakke" med klient/server, og huske å close socketen når man er ferdig.

Funksjonen har en evig loop, for å kommunisere med en klient, måten jeg har valgt å løse oppgaven på er å hardkode de første 2 meldingene mellom klient/server, så først mottar server ingenting, bare for at server kan sende tilbake ID noe som blir laget i main, og blir sendt med funksjonen. Hvis da bruker skriver 'Y' så fortsetter kommunikasjonen og man kan da sende så mange meldinger man vil til hverandre annenhver gang, hvis man ikke skriver Y, så valgte jeg å avslutte både server og klient (i terminalen). For at kommunikasjonen skal fortsette så bruker man read(), for å sende informasjon fra server så bruker jeg en while-loop som kjører så lenge det ikke har blitt tastet inn enter-knappen(\n) med getchar(); og lagrer så hver bokstav i et buffer som jeg sender med write(), for så å read() igjen, sånn kommuniserer server/klient med hverandre, helt til man trykker ctrl+c eller at klient ikke skriver 'y'

Klient filen:

I main() så henter jeg inn hvilken server(port) man skal koble seg opp imot, på samme måte som server-filen. Åpner en socket, å bruker 127.0.0.1(localhost) til loopback. Så connecter() man, og kjører funksjonen sin.

Funksjonen har en evig loop, som server. Og jeg har valgt å "hardkode" de første 2 meldingene imellom server/klient, den første sender en tom melding, som nevnt over. Den andre lagrer ID som man får tilsendt fra server'en og spør om man vil fortsette ved å skrive 'Y' eller noe annet for å avslutte. Hvis man fullfører handshaken så får man en melding om at handshake var successful. Ellers er den ganske lik som server, read() for å vente på svar fra server, write() for å sende meldinger, og huske å bzero() på bufferet for å fjerne det som var der.

Så for å kjøre server må man skrive `./main -port 6666 -id MittNavn` (eksempel portnr) og for å kjøre klient da må man skrive `./main -server 6666`
Hvis man skal bruke en port mindre enn 1024 så må man bruke "sudo" før `./main ..`

Til slutt vil jeg bare si at det har vært en bra eksamen, og jeg har lært masse underveis også :)