

upush_server.c

Serveren har syv hjelpemetoder som de fleste omhandler klient-strukturen på server-siden. I main venter den på at en klient skal sende input for deretter behandle dette. Når en klient sender input vil dette enten være registrering eller en lookup-melding. Serveren svarer deretter med en godkjenning av registrering eller gir informasjon om klienten som oppslag ble gjort på.

upush_client.c

Klientene har flere hjelpemetoder og noen av disse er delt med serveren som gjerne skulle vært lagt til i den egne filen for å slippe repeterende kode. I klient-filen finner vi structene Print, Message, PktAck og Blocked. Det første som skjer i main til klient er at den forsøker å registrere seg på serveren gitt fra parameter. Dersom dette ikke er mulig vil den avslutte programmet sitt med en gang, slik som presisert i oppgaveteksten. Det vil si at dersom det er pakketap ved første registrering, vil ikke klienten klare å kjøre. Dersom klienten får registrert seg går den inn i **hovedhendelsesløkken**. I **hovedhendelsesløkken** skjer hjerteslagfunksjonen slik som beskrevet i oppgaven. Den venter også på input fra tastatur eller melding fra nettet. Mot slutten av **hovedhendelsesløkken** vil programmet gå igjennom de eksisterende meldingene og sjekke dersom de har fått tatt imot en godkjenning av enten server eller andre klienter. Dersom ingen godkjenning er mottatt vil meldingene bli sendt opp til to ganger på ny, før programmet ikke gjør det lenger, det er denne biten som står for tapsgjenoppretting av meldinger på klienten sin side. Det er mulig for klienter å blokkere hverandre og oppførselen her stemmer overens med oppgavebeskrivelsen. Dersom klient blokkerer mottaker vil det ikke klient kunne skrive melding til mottaker, og mottaker sine meldinger vil ikke vises på skjermen hos klient. Prekoden set_loss_probability() trengte en liten vri, siden første kall alltid var 0 og fungerer nå. Klient kan avsluttes ved å skrive inn QUIT slik som beskrevet.

Felles for server og klient

Begge to deler samme klient-struct, men klient bruker flere av parameterene. Begge filene deler metode fra multiple.c. Begge filer har en **hovedhendelsesløkke**.

Valg

Jeg har ikke gjennomført taps-gjenoppretting på server sin side. Dette anser jeg ikke som et stort problem siden dersom klient ikke mottar acknowledgment fra server vil den prøve to nye ganger å sende lookup eller registreringsmelding, men samtidig var dette en del av oppgaven så det burde kanskje vært gjort. Det skrives ikke alltid ut en klar og tydelig feilmelding dersom formatet på en melding ikke er korrekt ved inntasting på klienten, dette kunne vært forbedret.

Problemer

Stop-and-await fungerer ikke helt slik som ønsket. Det viser seg at mot slutten av klient **hovedhendelsesløkken** at while-loopen sender meldinger for fort. Jeg har laget en egen Print struct som sørger for at duplikater av meldinger ikke skrives ut. Det ligger en del såkalte edge-cases som muligens ikke er tilfredsstillende optimalt slik som et ordentlig program burde kunne. Et annet problem som jeg gjerne skulle fanget opp er at klienter ikke avslutter av seg selv dersom de ikke lenger får kontakt med serveren, slik som oppgaven presiserte. Jeg har ikke gjort en sjekk om at alle tegn i meldinger er i ASCII-format, men har ikke funnet et problem med dette ennå. Serveren kjører for evig for øyeblikket, kunne vært ønskelig å avslutte denne på en sikker måte slik som klientene kan.