

## Hjemmeeksamen IN2140 Vår 2022

Programmene baserer seg rundt `nick_list` structen. Denne inneholder alt om en klient og hans meldinger. Denne structen inneholder igjen en lenkeliste meldinger som du kan se nærmere på `msg_list` structen. Det er en mismatch mellom det engelske og norske oppgavesettet i formatet av pakkene. Oppgaven følger det norske formatet.

**upush\_server.c**

Ingen leaks eller errors er oppdaget under testing for serveren, men de som har fulgt den engelske oppgave beskrivelsen vil ha et annet format på lookup og vil ikke kunne interface riktig med min server. Programmet starter opp og porten blir bindet til programmet hvis porten er tilgjengelig. Deretter starter hovedløkken og serveren lytter til input fra nettverk eller Admin, ved hjelp av `select` funksjonen.

Pakker fra nett er sendt til en funksjon som finner ut av innholdet til pakken, og en respons er sendt. Registreringsmeldingene vil forårsaket en allokering for å registrere. Dette fører til at hjerteslag meldinger fører til en allokering som vil bli umiddelbart frigjort. Se linje 56 – 57 i `upush_server` og `add_to_nick_list` funksjonen i `common`. Admin kan skrive QUIT for å avslutte eller LS for å se listen over klienter, Admin kan også bruke `ctr+c` for å avslutte på en trygg måte.

**upush\_client.c**

Programmet starter med å sette opp og kontakte serveren, hva som skjer her er ikke «viktig» for designdokumentet, og vil dermed ikke bli snakket om noe mer (se linje 383 – 433 for setup). Deretter starter hovedløkken. På lik linje så er det en `select` funksjon som overvåker noen fildeskriptorer. Disse fildeskriptorene er socketen, hjerteslag timeren, timeout timeren og STDIN. Programmet er designet på den måte at den skal sende meldinger asynkront. Hver klient man har snakket med har en meldingskø, hvor meldingene til klienten blir lagt til. Dette skjer når man skriver noe i terminalen. Teksten blir behandlet og lagt til i riktig kø, hvis det er aktuelt.

Når en hendelse skjer, og hendelsen bli behandlet så vil alle klientene loopes over og få sin første melding sendt (se første del av hovedløkken). Deretter vil klienten vente med å sende sin neste melding til en ACK fra mottakeren har kommet frem.

I situasjoner hvor meldingen ikke kommer frem eller ACKen ikke kommer frem så er det en tapsgjenoppretningsprotokoll på plass. Klienten vil sende på, søke opp informasjonen på nytt, og hvis informasjonen er ny så vil den sende to ganger til. Hvis informasjonen er ny så vil sekvensnumrene bli synkronisert.

Hjerteslagene er enkle registreringsmeldinger, som skjer hvert HJERTESLAG sekund (med mindre det ikke har blitt endret i `common` filen). Serveren tar et timestamp hver gang en pakke kommer og hvis en lookup skjer så sjekker serveren om sist sett var  $3 * \text{HJERTESLAG}$  sekund siden. Hvert HJERTESLAG sekund så sjekker serveren om noen har timet ut, og fjerner dem fra listen.

**common.c**

Denne fille inneholder funksjoner som brukes i både server og klient. Merkbart er `check_ascii`, `remove_spaces` og lenkelistefunksjonene. Lenkelistefunksjonene er ganske boiler plate, men det er de og stringfunksjonene er mest interessante.

`check_ascii` gir suksess når alle karakterer er mellom ASCII-verdi 32-126.