

Filene er organisert slik at:

1. De utdelte filene `send_packet.c` og `send_packet.h` skal være urørt
2. Filene `upush_client.c` og `upush_server.c` kompiles til kjørbare filer.
3. `Upush_client.c` og `upush_server.c` benytter seg av hver sin datastruktur i form av en lenkeliste. Denne datastrukturen og tilhørende funksjoner ligger i henholdsvis `datastructure_client.c` og `datastructure_server.c`, som kompiles sammen med `upush_client.c` og `upush_server.c`. Klient og server har hver sin header- fil, som inkluderes i begge `.c` filene.

Datastruktur:

Server benytter seg av en enkel lenkeliste som lagrer pekere til klienter i minnet. Klientene benytter seg av den samme lenkelista, men har i tillegg en ny lenkeliste i hver av klientene, hvor meldinger lagres mens de venter på å bli sendt, eller bli bekreftet. Ved trykk på `Ctrl + c` eller «QUIT» i terminalen vil klienten frigjøre minne og avslutte. Server vil frigjøre minne og avslutte ved `Ctrl + c`. Programmet har ingen valgrind- feil.

Klient:

**Registrering** skjer ved oppstart av klienten, og chat- tjenesten starter ikke før klienten har mottatt bekreftelse på at den er registrert hos server. Her avviker programmet fra oppgaven ved at klienten forsøker opptil 3 ganger, hvis ikke server svarer. Klienten går så inn i hovedloop, hvor den lytter til meldinger fra andre klienter samtidig med `stdin`, til den når **timeout**. Klienten kan kun ha 1 melding in- flight ad gangen, og lagrer nye meldinger i en kø til de kan sendes. Disse blir tatt ut av køen når klienten mottar bekreftelse. Det er mulig å **blokkere** og ikke- blokkere andre klienter, men dette fungerer kun så lenge klienten som skal blokkeres har sendt en melding til klienten som vil blokkere. Dette er en klønete måte å gjøre det på, ettersom det bør være mulig å blokkere andre klienter før en har mottatt en melding fra de. **Heartbeat** er ikke implementert. Dette fungerte halvveis inn i utviklingen, men det ble for mange feil tilknyttet det. **Oppslag** er blokkerende, og skjer derfor i hovedloop før timer er satt i gang. Etter å ha jobbet en stund med eksamen, ble det klarere hvordan alt egentlig burde vært gjort; i stedet for at noe blokkerer, burde alle innkommende meldinger blitt tatt hånd om av samme funksjon, som kunne delegert de ulike oppgavene videre. Da kunne blokkerende kall som oppslag vært ikke- blokkerende, og heartbeat hadde fungert. Det bør altså ikke være nødvendig at noe blokkerer i stor grad.

Server:

Server starter opp rett i hovedloop. Denne blokkerer helt til server får en forespørsel fra en klient. Da utfører den enten oppslag eller registrering, før den fortsetter i hovedloop.

Feil:

Det er ikke implementert en cache for siste meldinger fra klienter, som vil si at det kan forekomme at en melding skrives ut til `stdout` flere ganger.