

Tid: 16:00 - 20:00 | Arbeidsstasjon: 5 | Gruppemedlemmer: Jens  
Jahren Lauvrud, Lars Kristian Lie Hauge | Email:  
jensjl@stud.ntnu.no lkhaug@stud.ntnu.no

Vi tenker i utgangspunktet å skrive prosjektet i Rust. Det er flere grunner til det blant annet at det i utgangspunktet et språk vi har lyst til å lære fra før, at vi mener fokuset på minnesikkerhet, selv asynkront, gjenspeiler projektoppgaven, og at Rust fremstår mer relevant for oss fremover iom at vi begge går ELSYS.

Vi tenker å bruke UDP kommunikasjon i et Peer-to-Peer nettverk siden vi mener det gjenspeiler at heisene som et distribuert system og problemene framstilt i prosjektbeskrivelsen (f.eks. pakkeap, tap av strøm, osv.).

Heisene vil sende ut sitt verdensbilde og synkronisere den med de andre heisene ved å f.eks. gjøre bitvis OR mellom hver heis' calls. Siden lysene skal være en garanti tenker vi at minst to av heisene på nettverket må være enige og verifisere at calls er synkronisert, og at vi dermed har minst en backup, før vi kan skru på det tilhørende lyset.

For å hindre tap av ordre ved strømtap tenker vi at hver heis også skal kommunisere ut hva den mener alle heisene sine interne heisordre er og dette behandles på lik linje som ordinære ordre dog med en egen liste for hver heis. Om en heis mister strøm og starter opp igjen vil den dermed kopiere ordrene de to andre heisene mener at den skulle ha. Dette vil også gjelde om heisen mister sin interne tilstand av hvilken som helst grunn (f.eks. krasj, restart, osv.)

Om en av heisene detter av nettverket vil den operere som en singulær heis og operere som om den er eneste fungerende heisen derav ta alle daværende bestillinger. Pga. kravet av minst to synkroniserte heiser vil den ikke kunne skru på lysene, heisen vil derimot forsåvidt kunne fortsette å ta ordre.

For at en heis skal kunne markere en ordre som 'tatt' og at de andre heisene ikke umiddelbart skal ignorere den, da de gjør en OR mellom den innkommende 'tatt' og sin egen 'ikke tatt', må vi sjekke at heisen som mener å ha fullført ordren faktisk gjorde det. Dette kan gjøres ved å sammenlikne den gjeldene heisens tilstand (etasje, retning, osv.) med den ordren den mener å ha tatt og bare godkjenne benkjeden om alle variablene samsvarer. Altså at en om en heis i 2. etasje mener å ha tatt en ordre i 4. vil dette enkelt ignoreres.

Siden alle heisene kommer til å ha den samme koden vil avgjørelsen av hvilken heis som tar hvilken ordre bestemmes av den samme algoritmen og den vil være relativt enkel å endre underveis i prosjektet og vi har dermed ikke tatt en konkret avgjørelse av hvordan dette skal gjøres. Vi tenker at det derimot er hensiktsmessig å ta utgangspunkt i avstand og retningen hver heis bevrer seg i, samt at heisene helst vil distribuerer seg i ulike etasjer når de er uavirksom.