Kodefremføring (TTK4145)

# Forord

* Ikke endret heisdriveren

# timer.h / timer.c

* sjekker nåtid med satt timeout tid
* brukes for å time operasjoner uten å være blocking (non-blocking)

# elevator.h / elevator.c

* heisstrukturen
* elevator\_monitor: tråd som håndterer lokal-spesifikk heisfunksjonalitet
  + initialiserer heisen (lokalt)
  + setter lys og tar imot (lokale) knappetrykk
  + logger statusen 10 ganger i sekundet
* elevator\_initialize: initialiserer driver, finner (evt) en etasje, henter hierarkisk rolle
* (funksjoner som snakker for seg selv)
* should\_stop / should\_advance polles for å vite hva heisen skal gjøre
  + endeetasjer for redundancy (ikke gå utenfor grensene)

# filebackup.c

* henter og skriver lokal heis-status til fil

# common.h / common.c

* (selvforklarende funksjoner)
* common\_monitor: henter og sender bestillinger (til server)

# manager.h / manager.c

* MAX\_ELEVATORS: bufferstørrelse for buffer som holder informasjon om heisene i systemet
* remote\_elevator: buffer som nevnt
* manager\_lock: skrive-/leselås for buffer
* manager\_assign: finner passende heis for gitt bestilling

# main.c

* [main() på bunnen] start programmet i master- eller slavemodus (-m, -s [master IP])
* tcp\_server\_test: tråd for server-funksjonalitet
* common\_init
  + setter opp mutex-lås for lesing og skriving av felles bestillinger
* tcp\_client\_init
  + kobler til master eller går i «single elevator mode» om master ikke lar seg koble til
* setter opp tråder for felles bestillinger (common\_monitor) og heisen selv (elevator\_monitor)
* initialiserer heis
* setter opp polling med timer
* oppdaterer server når heisen endrer status
* kjører FSM for heisen så lenge ikke obstruction eller stop

# tcp\_server.c (tcp\_server.h kun funksjonshoder)

* bruker tråder for å håndtere klienter
* globale variabler:
  + socket\_desc: informasjon om socketen
  + client\_sock: midlertidig variabel for oppsett av ny klient
  + new\_sock: (midlertidig variabel) sender socketen til den nye clienten inn til behandler-tråden som peker
    - må bruke pekere som parameter i tråder?
  + server: serverens socket
  + client: (midlertidig) brukes under oppsett av ny klientkobling
  + remote\_elevator\_count: antall heiser koblet til
* tcp\_server\_init: setter opp socket for server-side
* tcp\_server\_test: tråd som tar imot nye klient-tilkoblinger
* elevator\_connection\_handler: tråd for hver klient som svarer på polling
  + «Ordbok»
    - b: button assignment
      * c: complete, bestilling utført
      * ellers: ny bestilling
      * svarer med hvilken heis som skal ta bestillingen
    - p: polling
      * r: rank, svarer med hvilken nummerering i heishierarkiet heisen har
      * a: assignments
        + svarer med bestillingsmatrisen
      * n: next master
        + svarer med ip-en til neste master
    - us: update state
      * oppdaterer heisens state, IDLE, MOVING, DOORS\_OPEN eller STOPPED
  + fjerner heisen fra remote\_elevators når den kobler fra
* add…/delete\_remote\_elevator
  + legger inn og sletter aktuelle heiser
  + rekalkulerer eierskap til alle bestillinger når en heis fjernes

# tcp\_client.h / tcp\_client.c

* DEFAULT\_PORT: TCP-porten som benyttes under kommunikasjon
* (mye selvforklarende)
* new\_master: sier ifra når «stasjonen» må skifte master
* lock: lås for kun en sending/mottak om gangen
* tcp\_client\_init: kobler til master-server og returnerer feil når den ikke klarer det
* tcp\_common\_call: sender bestilling og returnerer utvalgt heis
* tcp\_get\_station\_rank: spør hvilken rang/eierskap heisen har
* tcp\_get\_next\_master\_ip: (selvforklarende)
* tcp\_update\_status: sender ‘us’ update staus, heisens state, retning, etasje (max 999)
* tcp\_get\_common\_requests: oppdaterer «felles»-bestillinger
* tcp\_client\_send: sender melding til server og mottar/returnerer svaret