

# Designdokument - Eksamen IN2140

I dette designdokumentet vil jeg presentere designvalgene og koden for implementasjonen for følgende: TCP-koden i connection.c, utformingen av proxy.c og konverteringen mellom XML- og binær filer i recordFromFormat.c

## connection.c

I connection.c implementerer vi funksjoner som er ansvarlige for TCP-tilkobling og -håndtering. Følgende funksjoner er implementert:

- tcp\_create\_and\_listen: Oppretter en socket og venter på innkommende tilkoblinger.
- tcp\_accept: Forsøker å akseptere en innkommende tilkobling og returnerer den tilkoblede socketen.
- tcp\_read: Leser data fra en tilkoblet socket.
- tcp\_write: Skriver data til en tilkoblet socket.
- tcp\_close: Lukker en tilkoblet socket.

Implementasjonen av disse funksjonene følger standard TCP-tilkoblingsprosedyrer ved hjelp av systemkallene socket, bind, listen, accept, read, write og close. Feilbehandling er inkludert for å håndtere eventuelle feil som kan oppstå under TCP-operasjoner.

## recordFromFormat.c

I recordFromFormat.c implementerer vi funksjonene som er ansvarlige for konvertering av data fra XML- eller binært format til en Record-struct. Følgende funksjoner er implementert:

**1. Funksjonen "XMLtoRecord"** tar inn en buffer som inneholder et XML-objekt og konverterer det til en Record-struct. Denne funksjonen går gjennom hvert element (token) i XML-objektet ved hjelp av "strtok"-funksjonen. Den identifiserer hvert element basert på elementnavnet og setter tilsvarende verdier i Record-strukturen ved hjelp av hjelpefunksjoner som setSource, setDest, setUsername, setId, setGroup, setSemester, setGrade og setCourse. Til slutt returnerer funksjonen en peker til Record-strukturen.

**2. Funksjonen "BinaryToRecord"** tar inn en buffer som inneholder en binær streng og konverterer den til en Record-struct. Den leser flaggene fra bufferen for å identifisere hvilke felt som finnes i den binære strengen. Funksjonen tolker bufferen og setter tilsvarende verdier i Record-strukturen basert på flaggene. Denne funksjonen bruker også relevante hjelpefunksjoner som setSource, setDest, setUsername, setId, setGroup, setSemester, setGrade og setCourse for å sette verdiene i Record-structen. Til slutt returnerer funksjonen en peker til den konverterte Record-structen.

Implementasjonen av disse funksjonene avhenger av det spesifikke formatet og protokollen som brukes i dataene. Begge går gjennom bufferen som blir sendt inn.

## proxy.c

I proxy.c implementerer vi hovedprogrammet for proxy-serveren. Denne serveren tar imot tilkoblinger fra klientapplikasjonene "xmlSender", "binSender" og "anyReceiver", og håndterer datautveksling mellom klientene. Her er de viktigste funksjonene og deres roller:

- Forhåndsdefinerte variabler:
  - o Struct Client: inneholder informasjon om en tilkoblet klient.
  - o Typedef struct Client Client: definerer struct Client som en type Client
  - o Num\_clients: holder styr på antall tilkoblede klienter
  - o Struct Client \*client\_list[MAX\_CLIENTS]: Dette er en matrise av pekere til struct Client som representerer en liste over tilkoblede klienter. Størrelsen på matrisen er definert som MAX\_CLIENTS, som er 26.
- handleNewClient: Kalles når en ny tilkobling oppdages på socket-lytteren. Den aksepterer den nye tilkoblingen, oppretter en ny Client-struct og legger den til i klientlisten.
- removeClient: Kalles når en tilkobling blir brutt av en klient. Den frigjør ressursene til den tilkoblede klienten og oppdaterer klientlisten.
- forwardMessage: Kalles når proxyen mottar nok data fra en avsenderklient for å opprette en Record. Den bestemmer hvilken klient som skal motta Record-structen basert på destinasjonsfeltet, konverterer Record-meldingen til riktig format for klienten og sender den til klienten.
- handleClient: Kalles når aktivitet blir observert på en tilkoblet socket til en klient. Den håndterer mottak av data fra klienten, tolker formatet og oppretter en Record-struct ved hjelp av funksjonene i recordFromFormat.c. Hvis en klient er frakoblet, fjerner den klienten. Ellers videresender den Record-meldingen ved hjelp av forwardMessage-funksjonen.
- main: Hovedinngangspunktet for proxy-serveren. Den utfører initialisering av variabler og oppretter server-socketen. Deretter starter hendelsesløkken som venter på nye tilkoblinger, ny data på eksisterende tilkoblinger og hendelser som indikerer at en klient har koblet fra. Hendelsene blir håndtert ved å kalle de tilsvarende funksjonene (handleNewClient() og handleClient()). Hendelsesløkken fortsetter til ingen klienter er tilkoblet lenger. Til slutt lukkes server-socketen, og programmet avsluttes.

Programmet fungerte på alle testene utenom test-11-bin-split. Test-9-sleepzero får jeg ikke til å kjøre med RunTest.sh på ifi-maskinen via ssh. Kan også hende at det er noe feil i test-10-xml-split, i og med at brukernavnene er veldig lange. Tror kanskje applikasjonen ikke skjønner hvor Record-structene slutter, og hvor en ny starter.