

Øving: UDP, TLS og ASIO

Tomas Holt
26.01.2021

Frist 09.09.2021

Oppgave 1 - Kalkulator med UDP

Lag en enkel kalkulator-løsning hvor tjener/klient bruker UDP for kommunikasjon. Klienten sender tallene til tjeneren og får resultatet tilbake. Løsningen kan gjøres «lik» løsning gjort for øving med TCP-socket.

Oppgave 2 - TLS/SSL

Kjør TLS eksemplet på lenken fra leksjonen. Du vil nå kunne se vha. en «sniffer» at kommunikasjonen er kryptert.

Ettersom øvingen skal leveres i BB, beskriv/ta bilder av det du har gjort.

Frivillig oppgave (men anbefales!) - ASIO

Gå igjennom koden fra leksjonen eller evt. se på et annet rammeverk/språk for ASIO. Lag en enkel kommunikasjonsløsning hvor tjener returnerer teksten klienten sender til tjeneren. Her kan telnet/netcat godt brukes som klient...

Frivillig oppgave - Multicast

Lag en løsning hvor klienter kan registrere seg for å motta multicast-meldinger. Når en melding kommer så skal disse skrives ut for brukeren.

Lag så en «tjener» som sender ut meldinger som tjener-administrator selv skriver inn. Her blir altså meldinger/data kun sendt fra tjener til klientene. Slike løsninger er veldig aktuelle i forbindelse med spredning av multimedia-data.

Test løsningen med en tjener og gjerne flere klienter. **Merk at det kan være nødvendig å sitte på samme nett for at denne løsningen skal fungere, da det ikke er selvsagt at ruterne støtter/tillater multicast. Eduroam-nettet fungerer ikke i min test.**

Frivillig oppgave - Kalkulator med serialisering/DTO over UDP

«Utvid» løsningen over til å sende et egenlaget objekt som inneholder tallene (og gjerne også operator som skal brukes) som skal legges sammen på tjenersiden. Dette blir en løsning hvor man benytter seg av et Data Transfer Object (DTO). Det naturlige er da å lage en klasse som implementerer java.io.Serializable og som inneholder en liste av objekter som sendes til tjeneren.

Merk for å få dette til å fungere må man omforme det serialiserbare objektet til en byte-tabell. Dette blir en utfordring som dere bør klare å finne en løsning på selv.
