Journal for I4IKN øvelse 8 og øvelse 9

|  |  |
| --- | --- |
| Jeppe Benjaminsen | 201500154 |
| Søren Holm | 201409556 |
| Mikkel Poulsen | 20112893 |
| Morten Christensen | 201500162 |

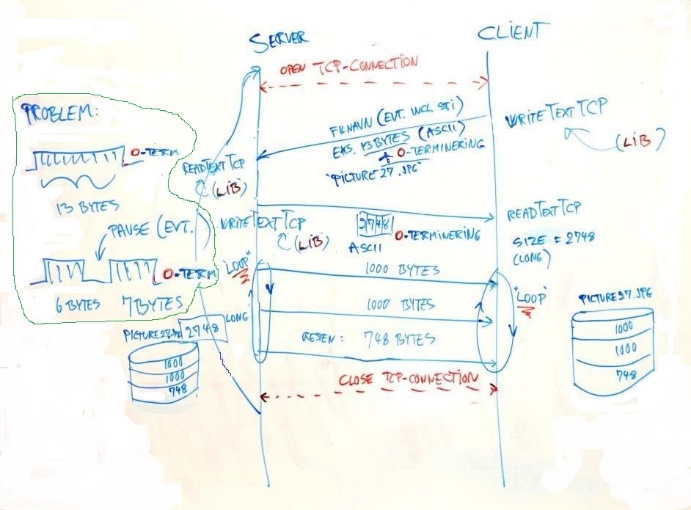
# Øvelse 8 – TCP/IP socket programmering

## Introduktion

I denne øvelse programmeres en socket til en TCP server og TCP client, og vi har valgt at skrive koden i C#. Det går ud på, at client skal kunne forbinde til server, og downloade en vilkårlig fil herfra. I det efterfølgende beskrives udviklingsforløb og funktionalitet.

## Udviklingforløb

Vi delte opgaven med at skrive client og server socket op, så vi havde en 2-mands-gruppe på hver opgave. Vi arbejdede iterativt ud fra skitsen på Figur 1, oppefra og ned, således at vi startede med at etablere TCP-connection, testede den. Dernæst forsøgte vi at requeste en fil, testede dette osv.



Figur 1 Protokol for TCP-overførsel

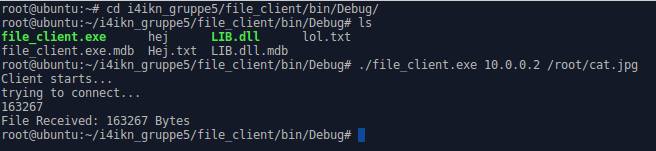
## Funktionalitet

Designet tager udgangspunkt i Figur 1. Når der er etableret en TCP-forbindelse mellem client og server, kan client requeste om at downloade en vilkårlig fil, enten med eller uden sti. Hvis der ikke er en sti angivet, vil serveren lede efter filen i sin root.

Hvis serveren finder filen, vil den returnere størrelsen af filen til client. På den måde kan client vide hvornår overførslen af filen er gennemført. Hvis den ikke finder filen vil den returnere 0, og client ved derfor at filen ikke kunne findes.

Hvis filen findes vil overførslen pbegyndes. Her sendes filen i masser af små pakker som er op til 1000 bytes stor.

På Figur 2 ses et eksempel på en test, hvor vi forsøger at overføre en fil cat.jpg. Det ses på konsoludskriften hvordan klienten først startes, derefter forsøger at oprette forbindelse og requester at hente cat.jpg. Filen findes, så serveren sender tilbage, at filen er 163267 bytes stor, hvorefter overførslen starter. Client udskriver så at filen er modtaget, og hvor mange bytes der blev modtaget.



Figur 2 – Overførsel af billede

# Øvelse 9 – UDP/IP socket programmering

## Introduktion

Der skal nu laves en UDP client og server, som vi vælger også at skrive i C#. Client skal kunne sende en forespørgsel til serveren, og der skal kunne understøttes én enkelt client. Der er mulighed for to forskellige forespørgsler: ”U” for at få information fra */proc/uptime*, og ”L” for */proc/loadavg.*

## Funktionalitet

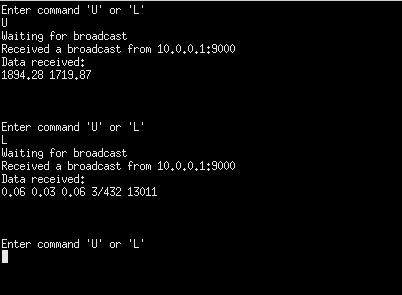
### Server

Serveren fungerer iterativt ved hele tiden at vente på forespørgsler fra en klient. Der modtages en forespørgsel som decodes, for at finde ud af hvad der skal sendes til klienten. Der åbnes en filestream til at læse */proc/loadavg* eller */proc/uptime.* Den information lægges i et bytearray, som sendes tilbage til klienten. Nu er serveren klar til at modtage en ny forespørgsel.

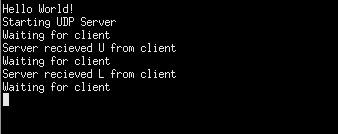
### Klient

Klienten kan sende en forespørgsel til en specifik server ved at indtaste serverens IP samt et ’U’ eller ’L’. I koden er porten hardcoded til 9000. Forespørgslen encodes til bytes for at kunne sende den. Når forespørgslen er sendt afsted, venter klienten på svar fra serveren.

### Eksempel på test

Nedenfor kan en test af systemet ses, hvor det ene billede er klient, og det andet er server.

Figur – UDP client



Figur – UDP server