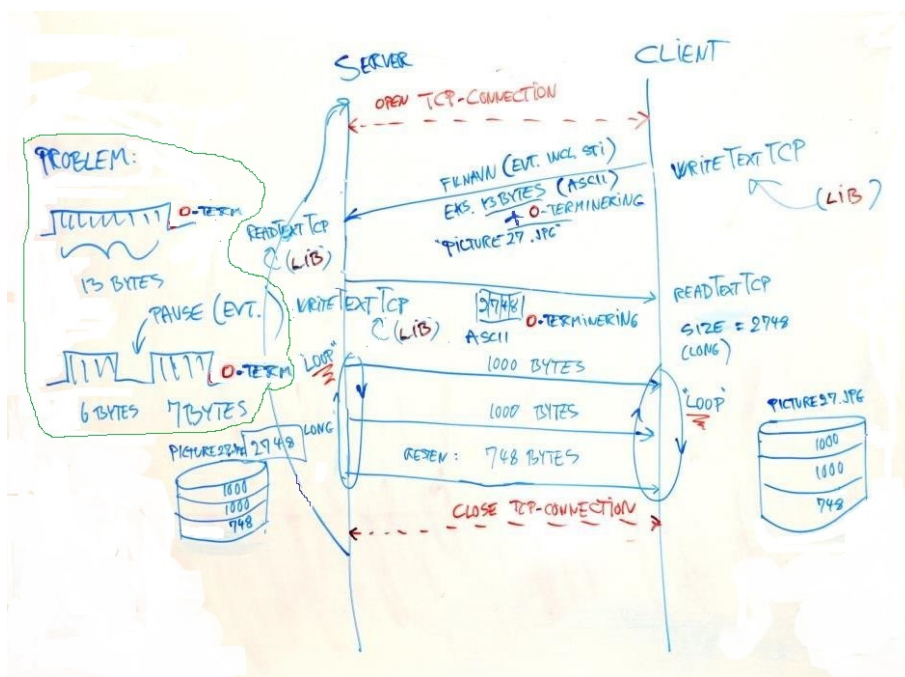


INGENIØRHØJSKOLEN AARHUS

IKN

E, IKT og EP



Øvelse 8 - Socket Programming

Udarbejdet af:

Simon Thrane Hansen

Lars Hjerrild

Kasper Lauge Madsen

201500150

201409555

201409873

Undervisere:

Torben Gregersen og

Lars Mortensen

28. september 2016

Indhold

1	Indledning	2
2	Udviklingsforløb	3
2.1	File-server	3
2.2	File-client	3
3	Test af TCP-server og client	4
4	Konklusion	6

1. Indledning

Denne opgave ophandler uarbejdning og test af en TCP-server og TCP-Client.

Serveren skal køre i en virtuel Linux-maskine og kunne supportere en client ad gangen. Serveren skal modtage en tekststreng fra clienten. Tekststrengen skal indeholde et filnavn, eventuel ledsaget af en stiangivelse. Tilsammen skal informationen i tekststrengen udpege en fil af en vilkårlig type/størrelse i serveren, som en tilsluttet client ønsker at hente fra serveren. Hvis filen ikke findes skal serveren returnere en fejlmelding til client'en. Hvis filen findes skal den overføres fra server til client i segmenter på 1000 bytes ad gangen – indtil filen er overført fuldstændigt. Serverens portnummer skal være 9000. Serveren skal desuden være iterativ og derfor være klar til at supportere en anden client efter en endt filoverførelse.

Clienten skal køre i en anden virtuel Linux-maskine. Denne client skal kunne hente en fil fra den ovenfor beskrevne server. Client'en sender indledningsvis en tekststreng, som er indtastet af operatøren, til serveren. Tekststrengen skal indeholde et filnavn + en eventuel stiangivelse til en fil i serveren. Client'en skal modtage den ønskede fil fejlfrit fra serveren – eller udskrive en fejlmelding hvis filen ikke findes i serveren.

2. Udviklingsforløb

2.1 File-server

I serveren blev, der implementeret en konstruktor og `sendFile`-metode.

Konstruktoren er lavet, så den opretter en ny `TCPListener`, der lytter på en vilkårlig IP-adresse på en hard-coded Port. Derefter opretter den en `TCPServer`. Serveren startes og venter på, at den opretter en TCP-connection med en client. Når forbindelsen med en client er oprettet den en `NetworkStream` til clientens socket. Serveren læser derefter, hvilken fil klienten ønsker overført. Derefter tjekker serveren om den ønskede fil findes og sender dette resultat til klienten. Hvis filen findes sendes denne til klienten ved at kalde metoden `sendFile`.

`sendFile` (`String fileName`, `long fileSize`, `NetworkStream io`) opretter et array af bytes med en faststørrelse for at overføre filen i stykker af 1000 bytes såfremt dette er muligt. Filen bliver læst ind ved brug af `FileStream` kommandoer og lagt i en variable, derefter overføres filen over `networkStreamen` i mindre stykker.

2.2 File-client

I klienten blev, der implementeret en konstruktor og `recieveFile`-metode. Konstrukturen tager to argumenter - et filnavn + stiangivelse og en ip-adresse til serveren. Det første, der sker i konstruktoren er, at der oprettes en ny `TCPClient`. Klienten forbinder derefter med en `TCP-Server` med den angivne IP-adresse og et hard-coded Portnummer. Klienten opretter derefter en `networkstream`, der skal tales med og sender en forespørgelse over at serveren skal sende den angivne fil til klienten. Klienten modtager derefter svar fra serveren om den angivne fil sendes på serveren. Hvis filens findes på serveren kalder klienten `recieveFile` og starter modtagelsen af filen. Efter filen er blevet overført lukker klienten forbindelsen til serveren.

`ReceiveFile` (`String fileName`, `NetworkStream io`, `long FileSize`)

3. Test af TCP-server og client

TCP-clienten og serveren blev testet ved at overføre diverse billeder og tekstfiler fra serveren på den ene virtuelle maskine til clienten den anden virtuelle maskine.

Nedenfor ses nogle eksempler på overførsler:

```
root@ubuntu:~/Desktop/IKN# ./Exercise8/file_client/bin/Debug/file_client.exe 10.0.0.1 /root/Desktop/car2.jpeg
Client starts...
Client started
root@ubuntu:~/Desktop/IKN# ./Exercise8/file_client/bin/Debug/file_client.exe 10.0.0.1 /root/Desktop/car3.jpeg
Client starts...
Client started
root@ubuntu:~/Desktop/IKN# ./Exercise8/file_client/bin/Debug/file_client.exe 10.0.0.1 /root/Desktop/abc.txt
Client starts...
Client started
root@ubuntu:~/Desktop/IKN# ./Exercise8/file_client/bin/Debug/file_client.exe 10.0.0.1 /root/Desktop/abc.txt
Client starts...
Client started
```

```
root@ubuntu:~/Desktop/IKN# ./Exercise8/file_server/bin/Debug/file_server.exe
Server starts...
>> Server Started
>> Data from client - /root/Desktop/car2.jpeg
>> Data from client - /root/Desktop/car3.jpeg
>> Data from client - /root/Desktop/abc.txt
>> Data from client - /root/Desktop/abc.txt
```

```
root@ubuntu:~/Desktop/IKN# diff -s /root/Desktop/car1.jpeg /root/Desktop/Compare
/car1.jpeg
Files /root/Desktop/car1.jpeg and /root/Desktop/Compare/car1.jpeg are identical
root@ubuntu:~/Desktop/IKN# diff -s /root/Desktop/car2.jpeg /root/Desktop/Compare
/car2.jpeg
Files /root/Desktop/car2.jpeg and /root/Desktop/Compare/car2.jpeg are identical
root@ubuntu:~/Desktop/IKN# diff -s /root/Desktop/car3.jpeg /root/Desktop/Compare
/car3.jpeg
Files /root/Desktop/car3.jpeg and /root/Desktop/Compare/car3.jpeg are identical
root@ubuntu:~/Desktop/IKN# diff -s /root/Desktop/abc.txt /root/Desktop/Compare/a
bc.txt
Files /root/Desktop/abc.txt and /root/Desktop/Compare/abc.txt are identical
```

4. Konklusion