

## **Journal for øvelserne 8 & 9**

|                          |                        |
|--------------------------|------------------------|
| Bjørn Nørgaard Sørensen  | Joachim Dam Andersen   |
| stud.nr: 201370248       | stud.nr: 201370031     |
| bjornnorgaard@post.au.dk | mr.anderson@post.au.dk |

October 23, 2015

## Indholdsfortegnelse

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Øvelse 8 - TCP/IP socket programming</b> | <b>1</b> |
| 1.1      | Introduktion . . . . .                      | 1        |
| 1.2      | Udviklingsforløb . . . . .                  | 1        |
| 1.3      | Funktionalitet . . . . .                    | 1        |
| 1.4      | Resultater . . . . .                        | 1        |
| 1.5      | Konklusion . . . . .                        | 1        |
| <b>2</b> | <b>Øvelse 9 - UDP/IP socket programming</b> | <b>2</b> |
| 2.1      | Introduktion . . . . .                      | 2        |
| 2.2      | Udviklingsforløb . . . . .                  | 2        |
| 2.3      | Funktionalitet . . . . .                    | 2        |
| 2.4      | Resultater . . . . .                        | 2        |
| 2.5      | Konklusion . . . . .                        | 2        |
| <b>A</b> | <b>Appendix: Sekvens diagrammer</b>         | <b>3</b> |

# 1 Øvelse 8 - TCP/IP socket programming

## 1.1 Introduktion

I denne øvelse i socket programmering, laves en TCP client og TCP server. Vi har valgt at programmere i det objektorienterede sprog C#. Clienten skal kunne forbinde til serveren, og downloade en fil herfra. Clienten og serveren køres på hver sin virtuelle linux maskine. I dette dokument beskrives udviklingsforløbet med tilhørende digrammer og kodeforklaringer.

## 1.2 Udviklingsforløb

Vi har designet koden således at server og client er opdelt i to hoveddele, hhv. constructor og overføringsfunktionalitet. For serveren betyder dette afsendingsfunktionalitet, og for clienten modtagelsesfunktionalitet.

Fizz1

Listing 1: derp

```
1 public FileServer()
2 {
3     //Setting up server and connecting client
4     //Getting filename an calculating lenght
5     //Sending file
6     //Closing connection
7 }
8 public void SendFile(string filename, long filesize, NetworkStream stream)
9 {
10    //Local variables
11    //Assigning variables
12    //Sending file
13    //Closing connection
14 }
```

## 1.3 Funktionalitet

## 1.4 Resultater

## 1.5 Konklusion

I arbejdet med TCP socket programmering er vi kommet frem til en læsning der opfylder kravene givet i opgaven. Det kan derfor konstateres at teorien stemmer overens med praksis.

## **2 Øvelse 9 - UDP/IP socket programming**

### **2.1 Introduktion**

### **2.2 Udviklingsforløb**

### **2.3 Funktionalitet**

### **2.4 Resultater**

### **2.5 Konklusion**

## A Appendix: Sekvens diagrammer

Figure 1: Sekvensdiagram for TCP server/client - uden Send og Receive metoder

