

# Praktikum Betriebssysteme: Projekt 1

## Client/Server Stream Socket

### Programmierung

Andreas Ruscheinski\*    Christian Delfs<sup>†</sup>    Fabienne Lambusch<sup>‡</sup>

24. Mai 2013

## Inhaltsverzeichnis

<b>1</b>	<b>Problem</b>	<b>2</b>
<b>2</b>	<b>Vorbetrachtung - Theorie</b>	<b>2</b>
2.1	Grundfunktionalitäten . . . . .	2
2.2	Kommunikation zwischen Server und Client . . . . .	3
2.3	Dateizugriff . . . . .	3
2.3.1	Beispieldateien . . . . .	4
2.4	Umzusetzende Funktionalitäten . . . . .	5
2.4.1	Administrator . . . . .	5
2.4.2	Normale Nutzer . . . . .	6
<b>3</b>	<b>Praktische Umsetzung</b>	<b>6</b>
3.1	Menü . . . . .	6
3.2	Client . . . . .	6
3.3	Server . . . . .	6
<b>4</b>	<b>Nutzerhandbuch</b>	<b>6</b>
4.1	Inbetriebnahme . . . . .	6
4.2	Server . . . . .	6
4.3	Client . . . . .	7
4.3.1	Login . . . . .	7
4.3.2	Studenten anlegen . . . . .	7
4.3.3	Gruppe anlegen . . . . .	7

---

\*Matr.-nr.: 211203494

<sup>†</sup>Matr.-nr.: 211204103

<sup>‡</sup>Matr.-nr.: ?????????

4.3.4	(Studenten)daten anzeigen . . . . .	7
4.3.5	Gruppe anzeigen . . . . .	8
4.3.6	Note hinzufügen . . . . .	8
4.3.7	Gruppenbesten ermitteln . . . . .	8
4.3.8	Gesamtbesten ermitteln . . . . .	8
4.3.9	Beenden . . . . .	8

## 1 Problem

Ziel des Projektes war es, zu lernen, wie Dateien mit Hilfe von Client/Server Stream Socket Programmierung (mit TCP/IP Sockets) angelegt und verwaltet werden können. Das wie folgt beschriebene Grundsystem wurde von uns mit einem Datei-Zugriffsschutz erweitert. Unsere Gruppe hat sich für Option b) entschieden: den Datei-Zugriffsschutz. Hierbei wird zwischen normalen Nutzern und Administratoren unterschieden, die sich durch verschiedene Rechte auszeichnen. Nach einem Login, der zum einen die Gültigkeit prüft, werden an dieser Stelle auch die Rechte ermittelt. Durch die Rechteverteilung werden dem Nutzer auf Clientseite verschiedene Funktionen zur Auswahl gestellt.

## 2 Vorbetrachtung - Theorie

### 2.1 Grundfunktionalitäten

Die grundlegenden Anforderungen an das Softwarprojekt sind wie folgt gegeben:

1. Speicherung der Studenten in separate Datensätze auf der Server-Seite. Enthaltene Informationen: Name, Matrikel-Nr., Geburtsdatum, Noten von n Lehrveranstaltungen.
2. Es gibt m Gruppen von Studenten, die jeweils mehrere Studenten umfassen ( $m \geq 3$ ).
3. Jede Gruppe repräsentiert einen Studiengang.
4. Die Datensätze jeder Gruppe werden in einer eigenen Datei gespeichert.
5. Gruppenbesten anhand des Durchschnitts ermitteln.
6. Mindestanforderungen: Mindestens drei Gruppen und in jeder: mindestens drei Studenten und jeder Studentendatei: mindestens drei Noten
7. Die Datensätze der Studenten sollten zugreifbar sein.
8. Bester aller Studenten soll ermittelbar sein.

## 2.2 Kommunikation zwischen Server und Client

Für unser Projekt verwenden wir verbindungsorientierte Sockets. Es wird eine festbestehende Verbindung zwischen dem Client und Server aufgebaut. Mit diesem Hilfsmittel wird ein Datenaustausch ermöglicht, der erlaubt, dass Daten zwischeneinander versendet werden können.

Der theoretische Ablauf, wie die Kommunikation erfolgt ist wie folgt: **Client-seitig:**

1. Socket erstellen
2. erstellten Socket mit der Server-Adresse verbinden, von welcher Daten angefordert werden sollen
3. senden und empfangen von Daten
4. evtl. Socket herunterfahren (shutdown())
5. Verbindung trennen, Socket schließen

**Server-seitig:**

1. Server-Socket erstellen
2. binden des Sockets an eine Adresse (Port), über welche Anfragen akzeptiert werden
3. auf Anfragen warten
4. Anfrage akzeptieren und damit ein neues Socket-Paar für diesen Client erstellen
5. bearbeiten der Client-Anfrage auf dem neuen Client-Socket
6. Client-Socket wieder schließen

## 2.3 Dateizugriff

In unserm Projekt nutzen wir das CDV-Dateiformat um die Datensätze von Studenten auf eine praktische Art und Weise speichern und auslesen zu können. Um die Studenten ihrer zugehörigen Gruppen zuzuordnen, werden diese in Verzeichnissen abgelegt.

Diese Verzeichnisse werden anhand ihrer Benennungen unterschieden, z.B.: steht der Ordner ITTI für die Studenten, die der Informationstechnik / Technische Informatik angehören.

Generell werden Studenten in einer Datei gespeichert, die als Bezeichnung die zugehörige Matrikelnummer besitzt. Innerhalb der Datei befinden sich die folgenden Datensätze, durch Semikolons von einander getrennt:

- Passwort
- Vorname

- Nachname
- Matrikelnummer
- zugehöriger Studiengang
- Geburtstag
- eine beliebige Anzahl an Noten

Der Datensatz eines Beispielstudenten sieht folgender Maßen aus:

mukitkarP;Max;Mustermann;113116119;Informatik;23.05.2013;1.0;1.3;2.7;5.0;1.0

Innerhalb des Programmes werden, bis auf die Noten, die Datensätze als Char-Arrays gespeichert. Wichtig ist es darauf zu achten, dass an letzter Stelle das Terminationszeichen ist. Das Passwort darf aus maximal 10 Zeichen bestehen. Für Vorname, Nachname und der zugehörige Studiengang sind 20+1 Char zur Verwendung unterstützt. Die Matrikelnummer besitzt 9 Stellen. Diese wird folglich in 9+1 Chars vom Client abgefragt. Ähnlich bei dem Geburtstag der in der Form DD.MM.YYYY in 11 Chars gespeichert wird.

Sowohl die einzelnen Noten, als auch die berechneten Durschnitte sind vom Datentyp Double. In die Datei werden die Noten als Chars der Länge 4 eingetragen, sodass sie immer die folgende Form besitzen X.Y wobei  $1 < X < 5$  und  $Y \in \{0,3,7\}$  ist.

### 2.3.1 Beispieldatein

Da zu Laufzeiten des Programms sowohl Gruppen als auch Studenten mit einer beliebigen Anzahl an Noten hinzugefügt werden, decken unsere Beispieldatein den geforderten Teil ab:

#### Gruppen:

- Informatik
- ITTI
- Elektrotechnik

#### Studenten der Informatik:

- 123;Tom;Klein;100000010;Informatik;01.01.1990;1.0;4.0;1.7;3.0
- 234;Max;Mustermann;100000011;Informatik;23.05.2001;1.0;1.3;2.7
- 345;Benjamin;Groß;100000012;Informatik;14.12.1986;1.0;1.3;2.7;3.3;1.7;5.0

#### Studenten der ITTI:

- 456;Even;Longer;100000013;ITTI;15.11.1977;1.3;2.7;3.3;1.7;5.0;1.0
- 567;Very;Long;100000014;ITTI;12.05.1993;1.0;1.3;2.7;1.0;5.0;1.0;1.0;1.0
- 678;Alfons;Hatler;100000015;ITTI;14.03.1989;1.0;1.0;1.0;1.0

#### **Studenten der Elektrotechnik:**

- 789;John;McClane;100000016;Elektrotechnik;24.12.1966;1.3;3.7;5.0;4.0
- 890;Hans;Gruber;100000017;Elektrotechnik;15.07.1991;1.3;5.0;2.7;1.0;4.0;1.7;3.0;2.0
- 901;Holly;McClane;100000018;Elektrotechnik;30.08.1988;1.7;1.3;1.3;1.7;1.0;1.7;1.7

## **2.4 Umzusetzende Funktionalitäten**

Neben den geforderten Funktionen, haben wir unser Programm um einige ausgewählte Funktionen bereichert. Unter Berücksichtigung der zwei Zugriffsarten mit unterschiedlichen Rechten werden hier die Funktionalitäten an zugehöriger Stelle beschrieben.

Dem Nutzer werden diese Möglichkeiten nach dem Login in einer Menüform zur Auswahl gestellt. Die Auswahl eines Menüpunktes erfragt auf Client-Seite, wenn nötig, Eingaben bzw. gibt die gewünschten Ergebnisse zurück.

### **2.4.1 Administrator**

**Studenten anlegen** Der Administrator muss bei dem Erstellen eines neuen Studenten dessen Vorname, Nachname, Matrikelnummer, Studiengang und Geburtstag angeben. Durch diese Eingabe wird eine Datei mit der Matrikelnummer als Name und im passenden Ordner (dem Studiengang) angelegt.

**Gruppe anlegen** Durch diese Funktion wird ein neues Verzeichnis erstellt, in das später Studenten eingefügt werden können.

**Studentendaten anzeigen** Um die Daten eines Studenten einzusehen, werden Kenntnisse über seinen Studiengang und seine Matrikelnummer vorausgesetzt. Zusätzlich wird auch dessen Durchschnitt angezeigt.

**Gruppe anzeigen** Nach der Eingabe des Studiengangs als Gruppennamen, werden alle Studenten aus diesem ausgegeben.

**Note hinzufügen** Damit einem Studenten eine Note hinzugefügt werden kann müssen Studiengang, Matrikelnummer und die hinzuzufügende Note als Eingabe angegeben werden.

**Gruppenbesten ermitteln** Für die ausgewählte Gruppe wird der beste Student anhand der einzelnen Durchschnitte ermittelt.

**Gesamtbesten ermitteln** Von allen Studenten, die gespeichert sind, wird der, mit dem besten Durchschnitt, ausgegeben.

**Beenden** Beendet die Kommunikation mit dem Server und schließt das Programm auf Seite des Clients.

#### 2.4.2 Normale Nutzer

**Daten anzeigen** Neben dem Vornamen, Nachnamen, Studiengang, Geburtstag und der Matrikelnummer wird der Durchschnitt ausgegeben.

**Beenden** Beendet die Kommunikation mit dem Server und schließt das Programm auf Seite des Clients.

### 3 Praktische Umsetzung

#### 3.1 Menü

Die in Kapitel 2.4 Beschriebenen Funktionalitäten führen zu zwei unterschiedlichen Menü-Formen. Abhängig von den Rechten, die dem Nutzer nach dem Login zugeteilt werden, wird ein entsprechendes Menü angezeigt.

```
Menü - Administrator
----
1)Student anlegen
2)Gruppe anlegen
3)Studenten anzeigen
4)Gruppe anzeigen
5)Note hinzufügen
6)Gruppenbesten ermitteln
7)Gesamtbesten ermitteln
8)Beenden
-----
Bitte Nummer eingeben:
>

Menü - Student:123456789
----
1)Daten anzeigen
2)Beenden
-----
Bitte Nummer eingeben:
>
```

## 3.2 Client

## 3.3 Server

# 4 Nutzerhandbuch

## 4.1 Inbetriebnahme

Als Betriebssystem wird eine Linux Distribution vorausgesetzt. Um das C Programm zu kompilieren, führt der Nutzer im Terminal „make install“ aus. Wichtig ist, dass er sich im richtigen Verzeichnis befindet - in dem sich auch das Skript „makefile“ befindet. In zwei verschiedenen Terminals werden nun die ausführbaren Dateien gestartet.

In Terminal 1 wird durch „./servern“ der Server gestartet.

Den Client startet der Nutzer in dem Terminal 2, durch die Eingabe von „./clientn 127.0.0.1“.

## 4.2 Server

Auf Seiten der Servers werden keine Eingaben oder Interaktionen zur Verfügung gestellt. Es werden gegebenenfalls Rückmeldungen ausgegeben, wenn der Client mit dem Server interagiert. Die Dateipflege kann neben dem Client-Interface auch manuell gehandhabt werden: Ordner als Studiengang hinzufügen, neue Dateien als Studenten anlegen oder Dateien öffnen und bearbeiten. Bei Einhaltung der oben genannten Spezifikationen kommt es dadurch zu keinen Fehlern.

## 4.3 Client

Bei Nutzung des Clients im Terminal 2 führt das Programm den Nutzer durch eine intuitive Menüführung. Zurück zu dem Hauptmenü kann der Nutzer jeder Zeit durch die Eingabe einer 0 gelangen.

### 4.3.1 Login

Nach Starten des Servers werden der Nutzernamen und das zugehörige Passwort abgefragt. Authentifizieren tun sich die Studenten durch ihre Matrikelnummer und das zugehörige Passwort. Die Administratoren besitzen den Nutzernamen „Admin“ und das Passwort „abcd“.

### 4.3.2 Studenten anlegen

Einen neuen Studenten anlegen ist nur für Administratoren möglich. Dieser wird aufgefordert stückchenweise die Stammdaten des Studenten einzugeben: Vorname, Nachname, Matrikelnummer, Studiengang und Geburtstag.

Hierbei wird der Ersteller auf die maximale Länge der Eingabe hingewiesen. Nachdem

alle Daten eingeben wurden müssen die Daten noch mit einer „1“ bestätigt werden. Hiernach wird durch diese Eingabe eine Datei mit der Matrikelnummer als Name und im passenden Ordner (dem Studiengang) angelegt.

#### **4.3.3 Gruppe anlegen**

Hier wird dem Administrator die Möglichkeit gegeben, bei Bedarf, einen neuen Studiengang hinzuzufügen. Auch hier wird ein Name mit der maximalen Länge von 20 Zeichen angefordert, sodass, nach der Bestätigung, ein neues Verzeichnis mit diesem Namen erstellt wird.

#### **4.3.4 (Studenten)daten anzeigen**

Je nachdem wer diese Funktion aufruft ereignen sich unterschiedliche Interaktionen.

Wenn jemand mit Administrator-Rechten sich Studentendaten anzeigen lassen möchte, wird er gefragt welchem Studiengang der Student angehört und welche Matrikelnummer dieser besitzt.

Handelt es sich um einen Studenten, ist keine Angabe seiner Matrikelnummer nötig. Es werden gleich seine Daten ausgegeben.

In beiden Fällen werden alle Daten und der zugehörige, bei Anfrage berechnete, Durchschnitt ausgegeben.

#### **4.3.5 Gruppe anzeigen**

Diese Funktion ist für Administratoren dahin gehend interessant, da sie für eine Gruppe alle Studenten ausgiebt, die sich in ihr befinden. Dazu ist es nötig den Namen des Studiengangs anzugeben.

#### **4.3.6 Note hinzufügen**

Für ein Software-Projekt, wie dieses, ist es sinnvoll, dass Studenten weitere Noten hinzugefügt werden können. Damit nicht jeder Student sich selbst Noten geben kann, ist dies nur vom Administrator aus möglich. Hierfür müssen wieder Studiengang und Matrikelnummer des Studenten angegeben werden. Dazu kommt naheliegender Weise auch die Note die dem Studenten zugedacht ist. Nach einer Bestätigung wird nun in der Studentendatei hinten die Note angefügt.

#### **4.3.7 Gruppenbesten ermitteln**

Diese Funktionalität ist ebenfalls den Administratoren vorbehalten. Es wird aus jedem Studiengang der Beste ermittelt. Dafür werden von allen Studenten die Durchschnitte gebildet und miteinander verglichen. Studenten wird nicht die Möglichkeit gegeben ihren Durchschnitt mit anderen zu vergleichen. Diese Funktion bleibt Administratoren vorbehalten.



#### **4.3.8 Gesamtbesten ermitteln**

Ähnlich wie bei dem Gruppenbesten werden auch hier Studenten verglichen. Um den Gesamtbesten zu ermitteln werden die Gruppenbesten anhand ihrer Durchschnitte verglichen. Nur Administratoren können den Besten ermitteln.

#### **4.3.9 Beenden**

Beendet die Kommunikation zwischen Client und Server.