

Aufgabenblatt 1

Aufgaben zur Socket-Programmierung

Verteilte Systeme
DHBW Karlsruhe
Stand Sommersemester 2018

Prof. Dr. Dirk Eisenbiegler

Aufgabe 1.1 - Time-Service

In dieser Aufgabe soll ein Internet-Service programmiert werden, von dem Clients das Datum und die Uhrzeit des Servers abfragen können.

Das Programm soll den Service auf dem Port 75 zur Verfügung stellen. Das Programm soll in einer Endlosschleife eine Verbindung nach der anderen entgegennehmen: sobald eine Verbindung beendet wurde, nimmt das Programm die nächste entgegen. Die Kommunikation zwischen Client und Server soll zeilenweise erfolgen. Sofort nach dem Verbindungsaufbau soll der Server dem Client die Textzeile "time service" zuschicken. Schickt der Client dem Server anschließend eine Textzeile mit dem Inhalt "date" oder "time" so antwortet der Server mit einer Textzeile, in der das Datum bzw. die Zeit steht. Sobald der Client eine andere Textzeile als "date" oder "time" zum Server schickt, beendet der Server die Verbindung.

Beispiel:

```
time service
date
15.1.2018
time
11:29:23
time
11:29:49
date
15.1.2018
end
```

- A) Schreiben Sie eine Klasse mit dem Namen TimeService, die diesen Dienst realisiert. Starten Sie die Klasse und testen Sie das Programm, indem Sie vom eigenen Rechner aus mit telnet eine Verbindung zu dem Service aufbauen.
Aufruf: telnet 127.0.0.1 75
Betrachten Sie dabei vor und nach dem Verbindungsaufbau die aktuellen Netzwerkverbindungen mit netstat.
- B) Stellen Sie von einem anderen Rechner aus eine Verbindung zu dem Rechner her, auf dem der Service läuft. Beobachten Sie dabei wieder die Netzwerkverbindungen mit netstat.
Hinweis: Beim Aufruf von telnet müssen Sie jetzt die Adresse des Zielrechners angeben. Die IP-Adresse eines Rechners erhalten Sie, indem Sie auf dem jeweiligen Rechner ipconfig aufrufen.
- C) Testen Sie nun das Verhalten des Programms, wenn Sie gleichzeitig mehrere Verbindungen zu dem Service herstellen.

Hinweise:

- x Zur Bestimmung der aktuellen Uhrzeit und des Datums können Sie die folgende Klasse verwenden:

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class Clock {

    private static SimpleDateFormat timeFormatter =
        new SimpleDateFormat("kk:mm:ss");
    private static SimpleDateFormat dateFormatter =
        new SimpleDateFormat("dd.MM.yyyy");

    public static String date() {
        return dateFormatter.format(new Date());
    }

    public static String time() {
        return timeFormatter.format(new Date());
    }

}
```

Aufgabe 1.2 - Client für Time-Service

Diese Aufgabe baut auf Aufgabe 1.1 auf. Es soll eine Client-Anwendung geschrieben werden, die über das Netzwerk Uhrzeit und Datum von einem Server abfragt. Auf dem Server läuft der Service Time-Service auf Port 75 (siehe Aufgabe 1.1).

Schreiben Sie eine Klasse mit dem Namen TimeServiceClient mit den beiden Klassenmethoden dateFromServer() und timeFromServer(). Beide Methoden haben je einen Parameter vom Typ String und einen Rückgabewert vom Typ String. Über den Parameter wird der Methode die IP-Adresse des Servers übergeben. Die Methoden sollen dann eine Verbindung zum Time-Service des Servers aufbauen, dessen Datum bzw. Uhrzeit abfragen und das Ergebnis als String zurückgeben.

Aufgabe 1.3 - Time-Service mit Multithreading

Diese Aufgabe baut auf Aufgabe 1.1 auf. Durch den Einsatz von Multithreading soll der Programmcode so geändert werden, dass mehrere Clients den Service gleichzeitig nutzen können.

Schreiben Sie eine neue Klasse `TimeServiceMultithreaded`, die den gleichen Service erbringt wie `TimeService`, jedoch mit dem Unterschied, dass zu dem Service mehrere Verbindungen gleichzeitig aufgebaut werden können. Dazu soll das Programm für jeden eingehenden Verbindungswunsch einen eigenen Thread starten, der diese Verbindung betreibt. Testen Sie das Programm, indem Sie mit `telnet` mehrere Verbindungen gleichzeitig aufbauen.

Tipp:

- × Wenn Sie das Programm im Debug-Modus starten, können Sie in Eclipse im Debug-View die einzelnen gestarteten Threads beobachten.