

5. Übung „Nebenläufige und verteilte Programmierung“ Abgabe am 5. Dezember, Besprechung am 6. Dezember

Aufgabe 1

In dieser Aufgabe sollen Sie einen Namensdienst implementieren. Dieser könnte dann zum Beispiel als Registrierungsservice für TCP-Dienste verwendet werden.

Implementieren Sie einen UDP-Server, welcher Schlüssel-Daten-Paare verwalten kann. Zur Vereinfachung verwenden wir sowohl für die Schlüssel, als auch für die Daten Strings. Klienten sollen neue Schlüssel-Daten-Paare einstellen, existierende löschen und die Daten zu einem Schlüssel nachschlagen können. Das Einstellen eines neuen Schlüssel-Daten-Paares soll natürlich nur dann möglich sein, wenn der Schlüssel noch nicht belegt ist.

Entwerfen Sie zunächst ein Protokoll, in dem Sie die Kommunikation spezifizieren. Implementieren Sie dieses dann unter Verwendung von UDP. Ihre Implementierung sollte auch beachten, dass bei UDP Pakete verloren gehen können. Zum Testen sollten Sie bei Ihrer Implementierung zufällig einige Pakete nicht schicken, da dieses Problem in lokalen Architekturen nur sehr selten oder gar nicht auftritt.

Aufgabe 2

In dieser Aufgabe sollen Sie einen einfachen Chat mit Hilfe von TCP implementieren. Der Chat soll aus einem Chatserver und beliebig vielen Chat-Klienten bestehen. Der Chatserver stellt seinen Dienst auf einem festen Rechner/Port zur Verfügung, der den Chatclients bekannt sein muss.

Folgende Funktionalitäten sollen zur Verfügung gestellt werden:

- Neue Klienten können unter Angabe eines Nickname in den Chat einsteigen, falls der Nickname von keinem bereits vorhandenen Klienten verwendet wird.
- Klienten können aus dem Chat aussteigen.
- Chatnachrichten der Klienten werden unter zusätzlicher Angabe des Nicknames bei allen Klienten angezeigt.
- Auch das Ein- bzw. Aussteigen von Klienten wird bei allen Klienten angezeigt.
- Der Chatserver soll robust sein. D.h. der Absturz eines Klienten soll vom Chatserver wie ein Aussteigen dieses Klienten behandelt werden.

Entwerfen Sie zunächst ein Protokoll, in dem Sie die Kommunikation spezifizieren. Implementieren Sie dieses dann mit `Sockets` bzw. `ServerSockets` und überlegen Sie, an welchen Stellen Fehler behandelt werden müssen.