

Aufgaben Socket Programmierung

1.1. Überprüfen Sie, wie sich die Port Nummer verhält, nachdem ein Client die Kommunikation mit dem Server aufgenommen hat.

Sie finden Hinweise dazu im Buch "TCP / IP Illustrated".

Was passiert mit dem Port, nachdem er von einem Client besetzt wird?

1.2. Lösen Sie die Selbsttest Aufgaben im Skript

Diese Lösungen dienen der Vertiefung des Stoffes und werden als bekannt voraus gesetzt.

1.3. Bauen Sie eine Socket Verbindung mit einem oder mehreren Kollegen auf

- versuchen Sie eine Verbindung im lokalen Netzwerk
- versuchen Sie eine Verbindung über den Proxy (Proxy befindet sich zwischen Client und Server)
- versuchen Sie eine Verbindung über die Firewall (Firewall befindet sich zwischen Client und Server)

Die folgenden Aufgaben sind etwas aufwendiger. Sie sollten in Teams zu maximal 3 Studenten gelöst werden.

Die Lösungen müssen auf den Server kopiert werden.

1.4. Groupware : Aufgabenbeschreibung

Simulieren Sie eine Produktionsstrasse mit Hilfe von Sockets:

- Benutzerteam 1 stellt eine Baugruppe her und gibt diese nach Fertigstellung an das nächste Team weiter
Das Herstellen der Baugruppe besteht im Erstellen von Grafiken und Dokumenten!
- Benutzerteam 2 stellt eine weitere Baugruppe her und montiert die Baugruppe 1 und die eigene Baugruppe zusammen.
Das Herstellen besteht im Erstellen von Ergänzungen zur ersten Grafik und Fortsetzung des Textes.

Die Kommunikation geschieht mit Hilfe von Sockets.

Am Schluss liegt ein Bild des Produktes vor, plus die Bedienungsanleitung (Dokument).

1.5. *GrafischeNetTools : Aufgabenbeschreibung*

Diese Aufgabe von mehreren Teams bearbeitet werden!

Wir haben einfache PortScanner, Whois Server, Daytime Server ... getestet.

Die Aufgabenteilung der Teams untereinander sieht wie folgt aus:

- das Leaderteam bestimmt die Architektur des Gesamtsystems
- die weiteren Teams erstellen spezielle Server, inklusive grafischer Oberfläche.
- die grafische Oberfläche muss in einem Look & Feel Guide (als Grafik oder wie auch immer) fest gehalten werden.
- die Funktionalität muss mindestens so gross sein, dass Sie alle Dienste, die wir im Unterricht untersucht haben, implementieren.

1.6. *Netzwerk Scanner*

Wir haben gesehen, dass es relativ leicht ist, *einen* Server zu scannen und dessen Dienste abzufragen.

Witzig wäre ein Scanner, der versucht festzustellen, welche Dienste in einem Netz angeboten werden, also auf *irgend* einem Server!

Wie könnte ein solcher Scanner aussehen?

Gibt es so etwas (auf dem WWW)? in Java? Wenn nicht in Java : wie könnte man ein solches Programm schreiben.

1.7. *Math-Server*

Schreiben Sie einen Server, der für Clients folgende Aufgaben erfüllen kann:

- der Client sendet dem Server eine mathematische Aufgabe
- der Server sendet die Lösung zurück

Schreiben Sie einen solchen Server. Beschränken Sie sich auf die Grundarithmetik, also: Integration und Differentialgleichungen braucht der Server nicht zu lösen.

1.8. *Proxy Server*

Schreiben Sie einen Proxy Server in Java.

Beachten Sie dabei:

- auf dem Web gibt es verschiedene Java Proxy Server
- Sie müssen aber in der Lage sein die Programme zu verstehen

2. Generell

Jedes Team stellt im Dezember die Lösung seiner gewählten Aufgabe vor (ca 10-15 Minuten).

Jedes Team erstellt ein Poster (1 PowerPoint Folie, auf der die Architektur der Lösung charakterisiert wird).