

Übungsblatt 1

Ausgabe: 23.10.2011

Abgabe: 04.11.2011

Aufgabe 1 (Webradio via TCP):

(16 Punkte)

Programmieren Sie ein Webradio. Verwenden Sie hierfür eine Client-Server-Architektur. Die Kommunikation soll zunächst über TCP-Sockets erfolgen, im Rahmen der nächsten Übungszettel jedoch auf UDP bzw. Multicast umgestellt werden. Bemühen Sie sich also um eine saubere Struktur Ihres Codes - Sie werden in den kommenden Wochen weiter mit ihm arbeiten müssen.

Ein Server bietet einen Audio-Stream an, der auf Anfrage an Clients geschickt wird. Clients sollen den Stream empfangen und abspielen können. Ferner sollen Clients Text-Nachrichten an den Server schicken können, die ebenfalls an alle angemeldeten Clients (aber nicht an den, der die Nachricht geschickt hat) übertragen werden.

Programmieren Sie zur Lösung dieses Übungszettels zunächst nur die Radio-Funktionalität.

Die Abgabe umfasst eine Dokumentation der verwendeten Konzepte, wichtigsten Design-Entscheidungen und hierfür relevantem(!) Code auf Papier, sowie ein ausführbares Jar in dem auch die Source-Files enthalten sein müssen. Das Jar benennen Sie bitte nach dem Schema: UB1_<Nachnamen der Tutanden>.jar.

- a) (4 Punkte) Laden Sie sich das Framework für die Aufgabe von der Veranstaltungsseite herunter. Benutzen Sie zum Lösen der Aufgaben die vorgegebene main-Methode, so dass alle Abgaben auf dieselbe Art aufgerufen werden können, sowie die vorgegebenen Interfaces. Das Manipulieren bestehender Interfaces ist nicht erlaubt. Ignorieren Sie für diese Übung das protobuf-package - dieses Paket wird für eine kommende Übung benötigt.

Machen Sie sich mit Sockets in Java vertraut.

Programmieren sie einen Server der mit einem `java.net.ServerSocket` eingehende Verbindungen akzeptiert und ein "Hello world." an den entfernten Socket sendet. Programmieren Sie einen Client der eine Verbindung mit einem solchen Server herstellt und eingehende Nachrichten auf der Kommandozeile ausgibt.

- b) (12 Punkte) Machen Sie sich mit der Java Sound API vertraut. Orientieren Sie sich an der vorgegebenen Beispiel-Klasse. Suchen Sie nach .wav Dateien die Sie abspielen können. Das abspielen dekodierter Formate, wie z.B. MP3, ist mit Java nicht ohne weiteres möglich (es sind externe Bibliotheken erforderlich). Optional können Sie es aber natürlich versuchen. Vergessene Sie nicht externe Bibliotheken im Jar zu inkludieren, sofern Sie solche verwenden

Der Server soll nun an neu angemeldete Clients einmalig das AudioFormat übertragen (Clients können davon ausgehen, dass sich das Audioformat nicht ändert) und anschließend eine Musikdatei auf den Stream schreiben. Clients sollen die Daten entsprechend verarbeiten. Es sollen sich annähernd beliebig viele Clients anmelden können.

Bis zum Beenden des Server-Programms soll die Musikdatei wiederholt abgespielt werden. Ferner soll die Möglichkeit bestehen, zur Laufzeit Server-seitig die abzuspielende Datei zu ändern. Optional kann auch eine Liste von Musikdateien verarbeitet werden.

Ermöglichen Sie die Steuerung zumindest des Servers über die Kommandozeile.