BAI4-RN	Praktikum Rechnernetze	HBN/SLZ
WS 2013	Aufgabe 2 – Entwicklung eines „Chat“ – Systems in Java	11.11./18.11.

Beschreibung des Chat-Systems

Das Chat-System ist sowohl eine Client/Server- als auch eine Peer-to-Peer Anwendung.
Der **Chat-Server** hat lediglich die Aufgabe, die Liste aller aktuell angemeldeten Chat-Clients (jeweils Chat-Name des Benutzers und Hostname des Clients) zu verwalten und allen Clients zur Verfügung zu stellen. Jeder neue **Chat-Client** muss daher für die Anmeldung eine TCP-Verbindung zum Chat-Server aufbauen, dem Chat-Server seinen Chat-Namen mitteilen (den Hostnamen des neuen Clients kann der Chat-Server aus dem Socket ermitteln) und regelmäßig die Liste aller angemeldeten Clients beim Chat-Server abfragen.
Der **Benutzer** des Chat-Clients kann nach Angabe des Chat-Server-Hostnamens und seines Chat-Namens beliebigen Text in ein Eingabefenster (Textfield) eingeben. Nach Abschluss einer Eingabezeile mit Return (newline) wird der eingegebene Zeilentext direkt (Peer-to-Peer) an alle anderen Chat-Clients geschickt (per UDP).
In einem zweiten Fenster werden alle Chat-Zeilen, die der Chat-Client von den anderen Teilnehmern erhält, angezeigt (Chat-Protokoll).

Chat-Server Spezifikation

Messages from Client to Server

`NEW <chatname>`
Der Client möchte sich unter dem angegeben Chat-Namen anmelden. Der Chat-Name darf keine Sonderzeichen und Leerzeichen enthalten!

`INFO`
Der Client fordert die aktuelle Liste aller aktiven Teilnehmer an.

`BYE`
Der Client meldet sich ab.

Messages from Server to Client

`OK`
Der Server hat den übergebenen Chat-Namen für den Client-Hostnamen in die Teilnehmerliste eingetragen. Die Client-Anmeldung ist somit korrekt erfolgt.

`LIST <n> Hostname-1 chatname-1 ... Hostname-n chatname-n`
Die Liste aller aktiven Teilnehmer (Clients) wird zurückgeliefert. Der erste Parameter n ist eine Zahl, die angibt, wieviele Paare (`Hostname chatname`) folgen.

`ERROR <reason>`
Die letzte Anfrage konnte aus dem mitgelieferten Grund nicht verarbeitet werden. Falls der Client angemeldet war, wird er sofort abgemeldet. Die TCP-Verbindung wird in jedem Fall vom Server geschlossen!

`BYE`
Client wird abgemeldet, TCP-Verbindung wird geschlossen.

Beispiel für eine korrekte Anwendung:

```
Client                          Server
NEW Superman -------------->  (Client-Hostname + “Superman” in die Teilnehmerliste eintragen)
             <--------------   OK
INFO         -------------->  (aktuelle Liste ermitteln)
             <--------------   LIST 2 lab30 Pussycat lab26 Superman
BYE          -------------->  (Client-Hostname + “Superman” aus der  Teilnehmerliste löschen)
             <--------------   BYE
```

Message-Name und Parameter werden durch mindestens ein Leerzeichen getrennt. Jede Nachricht muss mit einem `\n` (newline) abgeschlossen werden.

Server-Eigenschaften:
Der Server wartet auf TCP-Verbindungsanfragen auf Port 50000 und startet einen neuen Arbeits-Thread für jede Verbindung, über den dann das o.g. Protokoll abgewickelt werden kann.
Der Server benötigt keine graphische Oberfläche, sondern kann im Kommando-Fenster gestartet werden.

Anwendung im Praktikum:
Der von Ihnen geschriebene Chat-Server soll als zentraler Server von allen anderen Gruppen benutzt werden können, muss sich also exakt an das Protokoll halten. Der Server wird auf einem Rechner mit bekanntem Hostnamen gestartet.

Chat-Client Spezifikation

Der Client muss in einem anfänglichen Dialog den Hostnamen des Servers und den eigenen Chat-Namen vom Benutzer erfragen und sich anschließend sich mit dem Chat-Server über Port 50000 verbinden. Die Verbindung mit dem Server muss bis zur Abmeldung des Benutzers gehalten werden und wird dazu benutzt, regelmäßig (mind. alle 5 Sekunden) die aktuelle Teilnehmerliste anzufordern und anzuzeigen.

Jeder Chat-Client soll auf Port 50001 auf UDP-Pakete der anderen Teilnehmern warten und mit demselben UDP-Zielport eigene Textzeilen an alle Teilnehmer versenden.

Der Inhalt einer Textzeile ist eine Bytefolge mit folgender Struktur:

```
<Chat-Name>: <Textzeile><newline>
```

Beispiel: „`Superman: Gleich ist Wochenende – gehe nach Hause
`“
Ein Chat-Name darf maximal 20 Zeichen lang sein (Sonderzeichen und Leerzeichen sind nicht erlaubt). Eine Textzeile darf eine maximale Länge von 100 beliebigen Zeichen haben, als Zeichencodierung (Charset) ist „UTF-8“ zu verwenden.

Der Chat-Client soll eine einfache graphische Oberfläche haben:
- Eine Textarea (mit Rollbalken) für die angezeigten Beiträge (Chat-Protokoll)
- Ein Textfield für den eigenen eingegebenen Text (mit „Ende“-Button)
- Eine Textarea (oder ähnlich) für die Anzeige der Liste der angemeldeten Teilnehmer.

Berücksichtigen Sie die Folien aus der Vorlesung Betriebssysteme (Kap. 3 Folie 49-51)!

Hinweise

Der Chat-Client muss einerseits eine TCP-Verbindung zum Chat-Server aufbauen und halten, andererseits über UDP-Pakete mit den anderen Chat-Clients kommunizieren. Hierfür ist es sinnvoll, mehrere Threads zu erzeugen, z.B.:

- Thread A für die Kommunikation mit dem Server – nach der Initialisierung wird etwa alle 5 Sekunden die Teilnehmerliste abgefragt und die Anzeige aktualisiert
- Thread B für das Senden einer eigenen Textzeile an alle Teilnehmer (wird gestartet, nachdem eine Zeile fertiggestellt ist)
- Thread C für den Empfang aller Datagramme von den anderen Teilnehmern (lauscht beständig)