Richtlinien und Hinweise für die VS-Übungsaufgaben

- Jeder Student bekommt ein Projektverzeichnis unter /proj/i4vs/<loginname>.
- In diesem Projektverzeichnis muss für jede Übungsaufgabe ein Unterverzeichnis angelegt werden, z.B. für die fünfte Aufgabe mit:
  `mkdir /proj/i4vs/`*<loginname>*`/aufgabe5`
- Die Aufgaben müssen mit dem *abgabe*-Programm abgegeben werden. Um z.B. die fünfte Aufgabe abzugeben:
  `/proj/i4vs/pub/abgabe aufgabe5`
- Die Lösungen sollen in Gruppen erstellt werden sein.
- Wenn eine bestimmte Programmstruktur in der Aufgabenstellung verlangt wird, muss die Lösung diese einhalten.
- Es werden Punkte abgezogen, falls
  - verlangte Funktionalität nicht implementiert wird
  - nicht verlangte Funktionalität implementiert wird
  - das Programm nicht kompiliert.
- Der Code muss lesbar und nachvollziehbar sein. Falls komplizierte Teile vorkommen, sollten diese mit Kommentaren erläutert werden.
- Bei Problemen mit der Aufgabenstellung könnt ihr euch jederzeit an uns wenden:
  {felser, reiser}@informatik.uni-erlangen.de

Übungsaufgabe #1: Sun RPC und Java RMI

16.04.2003

In dieser Aufgabe sollen existierende RPC Systeme betrachtet werden. Dazu soll eine kleine Anwendung erstellt werden, welche Fernaufrufe verwendet. In dieser Übung wollen wir ein elektronisches schwarzes Brett erstellen. Ein Benutzer (Klient) soll dabei eine Nachricht mittels Fernaufruf an das Brett (Server) heften können. Die Nachricht besteht dabei aus einer User-ID (Integer), einem Titel (String) und der Nachricht (String). Des Weiteren soll ein Benutzer in der Lage sein, die letzten *n* (konfigurierbar, Integer) Nachrichten abzufragen.

Zum automatischen Testen soll ein einfacher Client implementiert werden. Dieser akzeptiert auf der Kommandozeile folgende Kommandos:

- *`board-client`* *servername* `POST` *uid title message*
- `board-client` *servername* `GET` *n*

Bei POST soll bei Erfolg die Ausgabe "OK" erfolgen, bei GET sollen die ermittelten *n* Nachrichten zeilenweise in der Form "*uid title message*" ausgegeben werden.

a) Sun RPC

Zunächst sollen SUN RPCs verwendet werden. Dazu muss u.a. die Schnittstellenbeschreibung `messageboard.x` erstellt werden.

Es ist ein Makefile zu erstellen, das bei Aufruf von "`make`" sowohl den Server als auch den Client erstellt. Die Programme müssen "`board-client`" und "`board-server`" genannt werden.

Hinweis: Der Client soll bei "GET" nur einen RPC an den Server schicken.

b) Java RMI

In dieser Teilaufgabe soll Java RMI zum Einsatz kommen. Der Server soll die gleiche Funktionalität wie in der vorherigen Teilaufgabe zur Verfügung stellen. Zusätzlich soll es möglich sein, dass ein Benutzer über neue Nachrichten informiert wird. Dazu muss ein Benutzer die Möglichkeit haben sich zu registrieren. Bei neuen Nachrichten soll er dann vom Server durch einen CallBack-Mechanismus informiert werden.

Der Client muss zusätzlich zu den bereits beschriebenen Kommandos ein Kommando `LISTEN` unterstützen, bei dem er via Fernaufruf einen Callback-Handler installiert und wartet. Bei jeder neuen Nachricht soll diese am Bildschirm ausgegeben werden, bis der Client mit Strg+C abgebrochen wird.

Es ist ebenfalls ein Makefile zu erstellen, welches die benötigen Class-Dateien erzeugt. Server bzw. Client sollten sich durch
"`java vs.boardserver`" bzw.
"`java vs.boardclient` *parameter*" starten lassen.

Abgabe: bis 29.04.2003 12:00 Uhr

(einzeln, keine Gruppenarbeit)