



Vernetzte Systeme

Übung 5

Ausgabe: **20. November 2002**

Abgabe: **2. Dezember 2002**

Bitte schreiben Sie immer Ihre(n) Namen auf die Lösungsblätter.

Vorbemerkung

Sie können sich den vorgegebenen Quellcode von unserer Homepage herunterladen. Da sich die Anwendung in mehrere Dateien unterteilt, haben wir für Oberon bzw. Java jeweils ein komprimiertes (zip) Archiv bereitgestellt. Da sich die Oberon- und Java-Version in Ihrem Aufbau und Verhalten teilweise unterscheiden, verweisen wir explizit auf die Kommentare im Quellcode.

1 Instant Messenger (14 Punkte)

Mit der Registrierung bei einem zentralen Server haben Sie in Übung 4 bereits einen wichtigen Teil unseres Instant Messengers geschrieben. In dieser Aufgabe geht es darum, die vom Server empfangene Benutzerliste zu verwenden, um anderen Benutzern Nachrichten zukommen zu lassen. Im Gegensatz zur Registrierung, bei der eine TCP-Verbindung verwendet wird, sollen diese Nachrichten als **UDP-Pakete** versendet werden.

Zur Lösung dieser Aufgabe müssen Sie sich mit dem gesamten Quellcode beschäftigen. Schauen Sie sich alle Module (Klassen) an, und machen Sie sich deren Bedeutung klar. Schauen Sie sich auch den `RegistrationClient` an, dessen Struktur und Verhalten sich leicht geändert haben!

Den bereits in der letzten Übung eingeführten Nachrichtentypen fügen wir noch den Typ `MESSAGE` (Wert `0x06`) hinzu. Dieser zeigt einem Empfänger an, dass eine Textnachricht im folgenden Format erhalten wurde¹:

Nachrichtentyp	Nachrichtenformat (Anzahl der Bytes)
MESSAGE	Typ(1) — Länge des Textes(2) — Text(max. 65535)

Lösen Sie nun die folgenden Teilaufgaben mit Hilfe der aufgeführten Hinweise und der Kommentare im Quellcode:

- a) (8 Punkte) Implementieren Sie die Prozedur `sendMessage` im Modul `MessageSender`.

Die Prozedur wird aufgerufen, wenn Sie eine Nachricht (*Instant Message*) versenden möchten. Schicken Sie dazu unter Verwendung eines UDP-Paketes den als Parameter angegebenen Text als Nachricht des Typs `MESSAGE` an den ebenfalls spezifizierten Benutzer. Versuchen Sie auch, mögliche Fehler zu erkennen und abzufangen.

- b) (6 Punkt) Implementieren Sie die Prozedur `receiveMessage` im Modul `MessageReceiver`.

Die Prozedur wird kontinuierlich in einem eigenen Prozess (Thread) durchlaufen und empfängt somit zeitlich unabhängig von der übrigen Anwendung UDP-Pakete anderer Benutzer. Schreiben Sie den notwendigen Code zum Empfang von Nachrichten des Typs `MESSAGE` über die vorhandene UDP-Verbindung, und zeigen Sie diese an. Versuchen Sie auch hier, mögliche Fehler zu erkennen und abzufangen.

¹Die maximale Länge des Textes von 65535 Bytes ist eher theoretischer Natur, da die tatsächlich mögliche Länge eines UDP-Paketes in Abhängigkeit der jeweiligen Protokoll-Implementierung variieren kann. Um diese Probleme zu umgehen, werden Pakete von Anwendungen meist nur mit einer maximalen Grösse von 512 Bytes versendet.

2 HTTP Status Codes (5 Punkte)

In der Vorlesung haben Sie erfahren, dass ein HTTP Server eine Anfrage mit verschiedenen Status Codes beantworten kann (Folie 2/19). Mittels der Anwendung `telnet` haben Sie bereits in der Übung 3 unsere Homepage heruntergeladen. Nun sollen Sie ausschliesslich verschiedene Status Codes erzeugen. Eine genaue Beschreibung dieser Codes und der Header-Zeilen finden Sie in den RFCs 1945 (HTTP 1.0) und 2616 (HTTP 1.1), z.B. unter <http://www.faqs.org>. Drucken Sie die von Ihnen erzeugten Ergebnisse jeweils aus, also etwa:

```
telnet dcg.ethz.ch 80
GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 15 Nov 2002 09:47:54 GMT
Server: Apache/1.3.23 (Unix) (Red-Hat/Linux) mod_ssl/2.8.7
       OpenSSL/0.9.6b DAV/1.0.3 PHP/4.1.2 mod_perl/1.26
X-Powered-By: PHP/4.1.2
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
[...]
```

- a) (1 Punkt) Versuchen Sie, den Status Code `404 Not Found` zu erzeugen.
- b) (1 Punkt) Versuchen Sie, den Status Code `304 Not Modified` zu erzeugen. Verwenden Sie dazu die Header-Zeile `If-modified-since`: gefolgt von einer Datumsangabe (`HTTP-date`), wie sie in den RFCs definiert ist.
- c) (3 Punkte) Versuchen Sie, möglichst viele (unterschiedliche) weitere Status Codes zu erzeugen. Spielen Sie dazu mit den verschiedenen Header-Zeilen herum, und probieren Sie auch fehlerhafte Angaben aus. Suchen Sie sich dazu beliebig viele HTTP-Server, da nicht unbedingt alle Codes von allen Servern erzeugt werden.