

Übungen zur Vorlesung  
**Systeme II / Netzwerke I**  
Sommer 2010  
Blatt 12

**AUFGABE 1:**

In dieser Aufgabe sollen Sie den einfachen "Echo"-Server von Blatt 10 zu einem kleinen Webserver erweitern, der eine HTML Seite zurückgibt.

1. Vervollständigen Sie das Codebeispiel von Blatt 10 so, dass ein Client mehrere Anfragen senden kann, ohne dass der Server sich beendet.
2. Vervollständigen Sie den Code so, dass Verbindungen auf Port 80 beantwortet werden. *Hinweis: Sie müssen nun das Java Programm mit Admin/Root Rechten ausführen, indem Sie unter unixartigen Systemen `sudo java Server` eingeben, wobei `Server` die `.Class` Datei Ihres Java Programms ist. Beenden Sie vorher andere Webserver, wie Apache, falls Sie diese installiert haben.*
3. Vervollständigen Sie den Code nun so, dass der Server einfache Browser Anfragen mit einer Meldung im HTML-Format beantwortet. Das heißt, senden Sie einem Browser folgenden Text als Antwort:

```
HTTP/1.1 200 ; OK\nContent-type: text/html\n\n<html>\n<head>\n<title>WebSever</title>\n</head>\n<body>\n<h1>It works!</h1>\n</body>\n</html>
```

4. Senden Sie statt des HTML-Codes ein Bild oder einen Film. Sie müssen also die Angabe des Content-Typs entsprechend ändern. *Hinweis: Ihr Browser ist in der Regel so konfiguriert, dass er beim Empfangen einer Musikdatei oder eines Films ein geeignetes Plugin zum Öffnen auswählt oder dies dem Betriebssystem überlässt.*

**AUFGABE 2:**

In den letzten Folien wurde der Begriff Proxy mehrmals erwähnt. Ein Proxy dient als Vermittler für Anfragen zwischen zwei Netzen, wenn diese z.B. unterschiedliche Protokolle verwenden. In dieser Aufgabe sollen Sie Ihren eigenen Proxy programmieren.

1. Vervollständigen Sie das Codebeispiel von Blatt 10 so, dass ein Client mehrere Anfragen senden kann, ohne dass der Server sich beendet.
2. Vervollständigen Sie den Code so, dass nur Verbindungen über einen bestimmten Port beantwortet werden, z.B. 1033.
3. Die Java Klasse `URL` verhält sich wie ein Webbrowser. Vervollständigen Sie den Code so, dass der Server den HTML-Code einer Webseite in der Konsole ausgibt. Ein neues `URL`-Objekt erzeugen Sie z.B. mit `new URL("http://www.google.de");`.
4. Vervollständigen Sie den Code nun so, dass der Server den HTML-Code an den Client zurückgibt und testen Sie das Ergebnis mit `telnet`.
5. Sie haben nun verschiedene Möglichkeiten den Proxy einzusetzen, z.B. über die Verbindungseinstellungen Ihres Betriebssystems. Benutzen Sie der Einfachheit halber Mozilla Firefox: Öffnen Sie in Firefox den Dialog "Firefox-Einstellungen". Wählen Sie unter "Verbindungen" den Button "Einstellungen". Wählen Sie nun "Manuelle Proxy-Konfiguration" und tragen Sie die Verbindungsdaten Ihres Proxy-Servers ein. Sollte Ihr Server als `localhost` laufen, entfernen Sie alle Einträge bei "Kein Proxy für". Starten Sie nun Ihren Proxy-Server und rufen Sie eine beliebige Seite auf. Klicken Sie auch auf Links in dieser Seite.
6. Überlegen Sie sich praktische Einsatzmöglichkeiten eines Proxys.
7. (Zusatz) Der Client sendet dem Proxy einen HTTP-Request in folgender Form

```
GET http://www.webseite.de/favicon.ico HTTP/1.1
```

Vervollständigen Sie den Code so, dass ein Browser den Inhalt von beliebigen Webadressen aufrufen kann.

### AUFGABE 3:

Analysieren Sie die folgenden Videostreams:

1. `http://movies.apple.com/media/us/html5/showcase/2010/demos/apple-html5-demo-tron_legacy-us-20100601_r848-2cie.mov`
2. `mms://c22033-o.w.core.cdn.streamfarm.net/22033mdr/ondemand/3087mdr/MDR_vgnwsm/FCMS-04ec9ba2-0133-459d-b183-73865cefaeef-7.wmv`
3. außerdem ein beliebiges Video von Youtube

Verwenden Sie dazu beispielsweise Wireshark oder `tcpdump`, um die Pakete zu analysieren. Erstellen Sie ein Koordinatensystem, wobei die X-Achse die vergangene Zeit in Sekunden angeben soll. Die Y-Achse soll hingegen die Sequenznummern der Pakete angeben.

Zeichnen Sie die nun für jede der oben genannten Video-Streams, die empfangenen und gesendeten Pakete in das Koordinaten dementsprechend ein.

Was können Sie aus den von ihnen erstellten Graphen folgern?