Dr. Matthias Grabert   Abteilung Angewandte Informationsverarbeitung   12. Juni 2002
Christian Ehrhardt   Blatt 7

Systemnahe Software (SS 2002)

Abgabetermin 18.6.2002

HTTP - Hypertext Transfer Protocol

HTTP ist das Protokoll, mit dem sich Webbrowser und Webserver unterhalten. In diesem Blatt soll ein kleiner Webserver implementiert werden, der einen kleinen Teil dieses Protokolls beherrscht.

Das Protokoll

Das HTTP-Protokoll wird über eine TCP-Verbindung abgewickelt. Der Server muß also auf ankommende TCP-Verbindungen auf einem von Euch wählbaren Port größer als 1024 warten. Das Protokoll selbst ist zeilenbasiert, wobei auch dieses Mal Zeilen wieder durch "\r\n" beendet werden. Allerdings sollte Euer Server so programmiert sein, daß er sowohl ein einzelnes "\n" alsauch "\r\n" als Zeilenende interpretiert.

Die Anfrage

Nach dem Aufbau der Verbindung beginnt der Client (also der Browser) die Kommunikation mit einer Anfrage. Diese Anfrage besteht aus mehreren Zeilen, und das Ende der Anfrage ist durch eine Leerzeile gekennzeichnet. Für unseren Server ist nur die erste Zeile von Interesse, diese muß aus 3 durch Leerzeichen getrennten Strings bestehen, die folgende Bedeutung haben:

- Ein Kommando, für unsere Zwecke muß dieses Kommando immer der Text "GET" sein.
- Der Pfad der Datei, die zurückgeliefert werden soll. Wenn dieser Pfad die Zeichenfolge ".." enthält soll Euer Server einen Fehler melden. Falls das nicht der Fall ist, soll die angegebene Datei unterhalb des Verzeichnisses /www/turing/htdocs/sai/ss02/soft/ gesucht werden. Falls der Pfad mit dem Zeichen "/" endet, soll noch der Text "index.html" angehängt werden. Wenn die so ermittelte Datei nicht existiert oder wenn es sich nicht um eine reguläre Datei handelt (fstat), so soll ein Fehler gemeldet werden. Beispiele:
  - Wenn als Pfad der Text "blatt2/" angegeben wurde, dann soll die Datei "`/www/turing/htdocs/sai/ss02/soft/blatt2/index.html`" verwendet werden.
  - Wenn als Pfad der Text "blatt2" (ohne "/" am Ende) angegeben wurde, dann soll ein Fehler erzeugt werden.
  - Ist als Pfad answers.html angegeben, so soll die Datei "`/www/turing/htdocs/sai/ss02/soft/answers.html`" verwendet werden.

  Was mit der Datei dann geschehen soll wird weiter unten erklärt.
- Die Version des Protokolls. Hier sollte der Text "HTTP/1.0" stehen. Dieses Feld kann für unsere Zwecke ignoriert werden.

Die Antwort

Die Antwort des Servers besteht aus einem Header gefolgt von weiterem Text. Header und Text werden durch eine Leerzeile getrennt. Die erste Zeile des Headers liefert ähnlich wie beim letzten Blatt einen Rückgabewert, der Aufschluß über den Erfolg oder den Grund des Mißerfolgs der Anfrage gibt. Diese Zeile hat die Form "HTTP/1.1 DDD mehr text\r\n".

DDD ist dabei ein dreistelliger Rückgabewert,

mehr text ist ein erklärender Text, der für unserem Zwecke "OK" im Erfolgsfall und "HTTP error" in allen anderen Fällen sein soll.

HTTP/1.1 soll einfach übernommen werden.

Die zweite Zeile des Headers sollte den Text "Content-Type: text/html" enthalten. Weitere Headerzeilen muß unser Webserver nicht erzeugen.
Nach dem Header und der Leerzeile folgt weiterer Text. Was hier genau kommen muß hängt vom Rückgabewert ab. **Anschließend beendet der Server die Verbindung, über eine Verbindung wird also immer nur eine Anfrage abgewickelt.** Der Webserver soll natürlich in der Lage sein, mehrere Verbindungen hintereinander zu bearbeiten.

Die Rückgabewerte

Im Erfolgsfall, d.h. wenn die als Pfad in der Anfrage angegebene Datei tatsächlich existiert, ist der Rückgabewert 200 zu verwenden. Der Text enthält dann den unveränderten Inhalt der Datei.
Falls ein Fehler auftrat, gibt der Rückgabewert den Grund für den Fehler an, der Text sollte den Fehler genauer beschreiben. Für unsere Zwecke genügt es immer folgenden Fehlertext zu verwenden, wobei **DDD** durch den Rückgabewert zu ersetzen ist (Newlines und Leerzeichen sind weitgehend irrelevant):

```
<HTML>
  <HEAD>
    <TITLE>DDD HTTP error</TITLE>
  </HEAD>
  <BODY>
    <H1>HTTP Error (DDD)</H1>
    Your request caused a HTTP Error (Code DDD).
  </BODY>
<HTML>
```

Unter anderem sind folgende Rückgabewerte vorgesehen (um Verwirrung beim Webbrowser zu vermeiden sollten keine hier nicht beschriebenen Rückgabewerte verwendet werden):

200 OK Die gewünschte Datei wurde gefunden (kein Fehler).

400 Bad Request Die Anfrage enthielt einen Syntaxfehler und konnte vom Server nicht bearbeitet werden.

403 Forbidden Der Server konnte oder wollte die Anfrage nicht beantworten, weil die nötigen Rechte fehlen. Die Anfrage wurde aber verstanden.

404 Not Found Die gewünschte Datei wurde nicht gefunden.

408 Request Timeout Der Client hat zwar eine Verbindung zum Server hergestellt, aber nie eine Anfrage geschickt.

414 Request-URI Too Long Der Pfad in der Anfrage war zu lang.

500 Internal Server Error Im Server ist ein Fehler aufgetreten. Diese Fehlermeldung kann von Eurem Server auch dann verwendet werden, wenn sonst keine hier beschriebene Meldung paßt.

501 Not Implemented Der Typ der Anfrage wird vom Server nicht implementiert.

503 Service Unavailable Der Server ist zur Zeit überlastet.

505 HTTP Version Not Supported Die vom Client angegebene Protokollversion wird vom Server nicht unterstützt.

Nicht alle diese Rückgabewerte müssen tatsächlich in Eurem Server verwendet werden.

Hinweise

- Die Funktionsweise des Servers kann "von Hand" mit Hilfe des Programms nc (siehe letztes Blatt) oder auch mit telnet (funktioniert genauso wie nc) getestet werden. Wenn alles klappt, sollte es möglich sein, den Server mit Netscape direkt anzusprechen. Dazu kann folgende URL verwendet werden: `http://rechner.mathematik.uni-ulm.de:4711/meinpfad` wobei rechner durch den Namen des Rechners auf dem der Server läuft zu ersetzten ist. Außerdem muß 4711 durch den von Euch gewählten Port ersetzt werden.
- Vor einem Test mit Netscape sollten unbedingt die Proxy-Einstellungen überprüft werden. Dazu bitte Edit->Preferences->Advanced->Proxies auswählen und dort den Knopf "Automatic Proxy configuration" anwählen und in das Texteingabefeld "`http://www.mathematik.uni-ulm.de/proxy.pac`" als "Configuration location" eintragen.
- Ports können belegt sein und ein Port bleibt nach dem Beenden Eures Servers unter Umständen noch für mehrere Minuten belegt. Es kann daher hilfreich sein, mehrere Ports durchzuprobieren bis ein nicht belegter gefunden wird.
- Unter Solaris muß beim Übersetzen von Programmen, die Sockets verwenden, die Option "-lsocket" mit angegeben werden.
- Interessante Manualseiten könnten unter anderem inet(7P) auf Solaris bzw. ip(7) auf Linux sein. Insbesondere der Abschnitt über struct sockaddr_in. Außerdem empfiehlt sich ein Blick auf die Manualseite byteorder(3). Falls wieder fdopen(3C) verwendet wird, könnte auch noch fwrite(3C) interessant sein.
- Wenn fdopen auf Sockets angewendet wird funktioniert der Mode "rw" nicht wie erwartet. Statt dessen kann z.B. "a+" verwendet werden. Grundsätzlich muß bei Sockets, die mit fdopen bearbeitet werden, zwischen einem lesenden und einem schreibenden Zugriff immer ein Aufruf von fflush stattfinden.