



## Aufbau von WWW-Servern

Dieser (und ein zweiter) Artikel beschreiben die Voraussetzungen und den Aufwand für den Aufbau eines WWW-Servers, die wichtigsten Installationsschritte, einige Anfangsschwierigkeiten und Fallen und geben Hinweise, wo Sie Hilfe und weitere Informationen finden können.

*Wozu braucht man einen WWW-Server? Welche Art von Informationen können dort bereitgestellt werden?*

Hier eine (unvollständige) Liste der Möglichkeiten als Anregung für eigene Ideen (Beispiel für ein wissenschaftliches Institut/eine Abteilung/eine Arbeitsgruppe):

- Vorstellung des eigenen Instituts (Lageplan, Verkehrsverbindungen (zur Orientierung für Gäste!), Forschungsgebiete, Spezialisierungsrichtungen der Einrichtung,
- Liste der Mitarbeiter mit Telefonnummern und E-Mail-Adressen, evtl. mit Verweis auf eigene Seiten der Mitarbeiter, auf denen sie sich und ihre Arbeit genauer vorstellen, Ansprechpartner für bestimmte Fragen/Verantwortlichkeiten (z.B. Studium, EDV etc., mit Sprechzeiten),
- Bekanntmachungen, Diskussionspapiere,
- Forschungsthemen, Liste der Veröffentlichungen, Ankündigungen/Terminplan/Agenda wissenschaftlicher Veranstaltungen,
- On-line-Katalog der eigenen Bibliothek, Liste weiterer fachspezifischer On-line-Kataloge (OPAC),
- Vorlesungsverzeichnis, Möglichkeit für Studenten, sich interaktiv ihren eigenen Stundenplan zusammenzustellen (siehe <http://www.educat.hu-berlin.de/cgi-bin/htmperl/~judy/studbera/frage1.html>), Prüfungstermine etc.
- Zusammenstellung von „Fundorten“ fachspezifischer Informationen im Internet.

Weitere Ideen finden Sie bei einem „Besuch“ auf den schon existierenden Servern der HU (zu erreichen über den URL=<http://www.hu-berlin.de/inside/hu-www-serv.html>) oder auf Servern anderer wissenschaftlicher Einrichtungen im Internet.

Summa summarum schafft der Aufbau eines WWW-Servers eine weitere Möglichkeit, die eigene Arbeit und Leistungsfähigkeit vorzustellen und Kontakt zu Menschen zu finden, die an gleichen Themen arbeiten oder Ihre (Dienst-) Leistungen in Anspruch nehmen wollen.

### Technische Voraussetzungen

Server-Software gibt es für alle verbreiteten Hardware-/Betriebssystem-Plattformen (UNIX, Linux, VMS, Windows95/Windows NT, NeXTStep, OS/2, Macintosh, sogar für DOS und Windows [1]. Darunter ist sowohl freie (kostenlos bei nichtkommerzieller Nutzung) und kommerzielle Software zu finden. Als Plattformen für hinreichend belastbare WWW-Server, die auch grundlegenden Sicherheitsanforderungen genügen sollen, kommen jedoch nur echte Multitasking-Systeme (z.B. UNIX,

aber nicht DOS oder Windows) in Frage. Kommerzielle Server-Software wird z.B. von Netscape Communications Corp. (Netsite Communications Server, Netsite Commerce Server, für UNIX, siehe [http://home.mcom.com/MCOM/products\\_docs/server.html](http://home.mcom.com/MCOM/products_docs/server.html)) und von O'Reilly Inc. (WebSite, für Windows95/Windows NT, siehe <http://clubweb.ora.com/news/facts.html>) entwickelt. Das Interesse an der kommerziellen Nutzung des Web ist groß - die Gründung von Commerce Net (siehe <http://www.commerce.com/>), die große Breite von Angeboten (von Werbung über Reiseangebote, Internet Bookshops bis zum Pizza-Service) und die Tatsache, daß auf der 2. und 3. Internationalen WWW-Konferenz [4] bereits die Hälfte der Teilnehmer aus dem kommerziellen Bereich kam, belegen diese Tatsache. Solche Anwendungen verlangen die Implementierung erweiterter Funktionalitäten in den Servern und Clients (z.B. für gesicherte (verschlüsselte) Transaktionen, für die Autorisierung/Identifizierung von Benutzern usw.).

### Hardware/Betriebssystem

WWW-Server können bereits auf einfacher und preiswerter Hardware aufgebaut werden. Der Server der Abteilung Pädagogik und Informatik (Institut für Wirtschafts- und Erwachsenenpädagogik) lief lange Zeit sogar auf einem PC 386SX/25 mit 8 MB RAM und 120-MB-IDE-Festplatte unter Linux. Die Anforderungen unterscheiden sich natürlich danach, ob der Server hauptsächlich bereichsinternen Zwecken dienen soll oder wegen seines Inhaltes weithin im Internet bekannt ist und viele Benutzeranforderungen zu bedienen hat. Weiterhin ist zu beachten, ob der Rechner nur als WWW-Server betrieben wird oder weitere Anwendungen darauf laufen sollen. Die Mindestausstattung für einen kleinen, wenig belasteten Server unter Linux wäre ein PC 386DX/40 mit 8 MB RAM und einer 150-MB-Festplatte. Ist im Bereich ohnehin eine Workstation vorhanden, die ständig in Betrieb ist, kann der Server dort als weitere Anwendung mitlaufen, sofern die Maschine nicht mit ihren sonstigen Aufgaben bereits ausgelastet ist oder bestimmte Sicherheitsanforderungen das verbieten.

### Server-Software

In diesen Artikeln beschränke ich mich auf die Beschreibung kostenlos (bei nichtkommerzieller Nutzung) verfügbarer Software, v.a. auf die vom CERN („CERN-WWW-Server“, CERN httpd) [7] und vom NCSA („NCSA-WWW-Server“, NCSA httpd) [12] entwickelten Server. Per anonymous FTP (auch direkt mit einem WWW-Client) sind sowohl der Quelltext als auch einige fertig übersetzte Versionen (Binaries) für viele UNIX-Plattformen zu

bekommen [9],[13].

### Binaries für WWW-Server (Auswahl):

**CERN httpd** (aktuelle Version 3.0):

für IRIX 5.2, SunOS 4.1.3, Solaris 2.3, HP-UX 9.0, AIX 3.2, OSF/1 3.0, Linux 1.2.8, SCO 3.2, NeXTStep 3.2.

**NCSA httpd** (aktuelle Version 1.4):

für IRIX 5.2, SunOS 4.1.3, Solaris 2.4, AIX 3.2.5, HP-UX 9.05, OSF/1 3.0, Linux 1.2.8.

### **Dokumentation, Unterstützung**

Für beide Server gibt es sowohl Dokumentationen in HTML als auch Handbücher im PostScript-Format [10],[14] - jeweils in Englisch. Die HTML-Dokumentationen kann man entweder direkt übers Netz lesen - diese repräsentieren natürlich den aktuellen Stand und werden durch FAQs, Beta-Test-Berichte usw. ergänzt - oder als tar -Archiv per FTP [10],[14] herunterladen, um sie lokal zu installieren. Wenn Sie für Ihren Client einen Proxy Server (siehe Teil 2) konfigurieren (z.B. <http://www.huberlin.de/>), befinden sich die Seiten der Online-Dokumentation vom CERN und NCSA (zumindest teilweise) mit ziemlicher Sicherheit im Cache, so daß sich die Zugriffszeiten deutlich verkürzen. Weitere Informationsquellen sind die entsprechenden News-Gruppen, v.a.

comp.infosystems.www.announce  
comp.infosystems.www.providers  
comp.infosystems.www.users  
comp.infosystems.www.misc  
de.comp.infosystems

und nicht zuletzt auch unsere Mailing-Listen [5]:

[www-adm@rz.hu-berlin.de](mailto:www-adm@rz.hu-berlin.de) und  
[www-autoren@rz.hu-berlin.de](mailto:www-autoren@rz.hu-berlin.de)

(keine listserv-Listen (!), wer eingetragen werden möchte, bitte Mail an [w.naumann@rz.hu-berlin.de](mailto:w.naumann@rz.hu-berlin.de) schicken).

### **Unterstützung durch das Rechenzentrum**

- a) Wir leisten Hilfe beim Aufbau eigener WWW-Server. Ansprechpartner ist Herr W.Naumann (Tel. 2093-2602, E-Mail an [webadm@rz.hu-berlin.de](mailto:webadm@rz.hu-berlin.de)).
- b) Einrichtungen der HU, die den Aufwand zum Aufbau und Betrieb eines eigenen WWW-Servers nicht leisten können, stellt das RZ seinen Server `www2` zur Verfügung. Da der Rechner in technischer Hinsicht vom RZ betreut wird, brauchen Sie sich nur um die Herstellung Ihrer HTML-Dokumente zu kümmern. Bitte wenden Sie sich an Herrn Ohst (Tel. 2093-2402, E-Mail an [webadm@rz.hu-berlin.de](mailto:webadm@rz.hu-berlin.de))

### **Erforderliche Kenntnisse**

- Es sei hier deutlich darauf hingewiesen: Zur Installation eines WWW-Servers (auf UNIX-Maschinen) braucht man vor allem Kenntnisse als System- und Netzadmini-

strator und Zugriffsrechte für Root - ersteres weniger für die Installation der Software als vielmehr für die richtige Einbindung des Servers in das Gesamtsystem und die Abschätzung der Auswirkungen auf die Systemsicherheit, die Performance des Systems usw.! Das sollte nicht unterschätzt werden, da weder in diesem Artikel noch in irgendeiner Dokumentation eine vollständige Liste aller „Gefahrenstellen“ enthalten sein kann. Dazu gehört auch, daß neuentdeckte Sicherheitslücken meist nur im Systemverwalter-Umfeld (und in Hacker-Kreisen) bekannt werden.

- Für die Verwaltung und Aktualisierung der HTML-Dokumente, die der Server zur Verfügung stellen soll, reichen in der Regel einfache UNIX-Kenntnisse auf Nutzer-ebene (Ausnahmen: eigene und fremde Prozeduren/Programme, die das Common Gateway Interface (CGI) zur Herstellung dynamischer Dokumente nutzen).

- Zur Erstellung von HTML-Dokumenten muß man im einfachsten Fall mit einem ASCII-Editor eines beliebigen Betriebssystems (edit unter DOS, vi unter UNIX, ...) umgehen können - das ist möglich, aber unbequem - oder einen der mehr oder weniger komfortablen/leistungsfähigen HTML-Editoren bzw. -Konverter bedienen können (siehe u.a. den Artikel „Werkzeuge für HTML“ in diesem Heft und [6]) und natürlich Kenntnisse der WWW-eigenen Sprache HTML besitzen.

### **Grundlagen**

Das Programm, das bisher als ‘WWW-Server’ bezeichnet wurde, ist ein Prozeß, der so wie andere Kommunikationsprogramme auf UNIX-Rechnern (ftpd, telnetd usw.) unauffällig im Hintergrund läuft, als *Dämon*. Da die Kommunikation mit den Clients (z.B. NCSA Mosaic) auf der Grundlage des HTTP-Protokolls erfolgt, heißt er HTTP-Dämon, *httpd*. Er wird entweder beim Hochfahren des Systems gestartet und bleibt ständig aktiv, oder er wird bei Anforderung eines HTML-Dokuments vom übergeordneten Dämon `inetd` ‘geweckt’, bedient die Anforderung und begibt sich wieder zur ‘Ruhe’. Die Anfragen müssen auf einem bestimmten *Port* eintreffen (der Standard-Port für das HTTP-Protokoll ist Port 80). Ports mit Nummern <1024 sind privilegierte Ports, weil Prozesse, die Anfragen auf diesen Ports bedienen, nur von Root gestartet werden können. Das bietet dem Nutzer eine gewisse Sicherheit, daß die ihm antwortenden Prozesse auch die sind, die er erwartet. Prozesse auf Portnummern >1024 können von jedem Nutzer gestartet werden, so daß dort u.U. auch Programme (trojanische Pferde) anzutreffen sind, die unter falschem Namen unerwartete (meist schädliche) Dinge tun (Sicherheitslücke!). Der HTTP-Dämon, der von Root gestartet wurde, wechselt nach dem Start - bevor er Nutzeranfragen beantwortet - auf eine andere UserId, damit Zugriffe auf das lokale Dateisystem nur mit minimalen Rechten erfolgen (Voreinstel-

lung UserId nobody, GroupId nogroup - in der Konfigurationsdatei vornehmbar).

## Installation

Ich beschränke mich auf einen Überblick über die wichtigsten Installationsschritte und einige Schwerpunkte bei der Konfiguration des Servers am Beispiel des CERN-WWW-Servers (Version 3.0), der auch auf den Servern im RZ installiert ist. Die Dokumentation sollten Sie also trotzdem unbedingt lesen.

### Beschaffung der Software

Die Server-Software liegt auf FTP-Servern als gepacktes tar-Archiv [11],[15]. Sofern es sich um eine vorcompilierte Version für eine bestimmte Plattform handelt, enthält das Archiv sowohl die ausführbaren Programme (httpd, htadm usw.) als auch Beispiele für Konfigurationsdateien. Nach dem Auspacken des Archivs in einem geeigneten Verzeichnis, z.B. /usr/local/cern\_httpd, gibt es dort die Unterverzeichnisse config, bin, cgi-bin.

### Konfigurationsdatei

Die Funktionen des Servers werden durch Eintragungen in seiner Konfigurationsdatei eingestellt. Beim Start sucht er nach der Datei /etc/httpd.conf. Über die Option -r kann eine andere Konfigurationsdatei benannt werden: `httpd -r <config-filename>`. Das Verzeichnis config und die Dokumentation [10],[14] enthalten Beispiel-Konfigurationen für verschiedene Fälle.

Für die Erstinstallation eignet sich am besten das Beispiel config/httpd.conf. Man erzeugt einen symbolischen Link nach /etc/httpd.conf und ändert einige Zeilen nach den eigenen Erfordernissen:

- **ServerRoot:**

Das Verzeichnis, in dem die Software ausgepackt wurde.

- **Port:**

Ein Port >1024, falls Sie zu den Server nicht unter Root starten wollen/können (nur zu Testzwecken!).

- **UserId / GroupId:**

Voreingestellt auf nobody/nogroup. Nutzer/Gruppe mit minimalen Zugriffsrechten. Überprüfen Sie deren Existenz in /etc/passwd bzw. /etc/group.

- **AccessLog, ErrorLog:**

Unbedingt angeben! Gerade über die Logdateien lassen sich Konfigurationsfehler leichter finden. Später - bei mehr Erfahrung - kann man das Logging auf ausgewählte Ereignisse einschränken.

- **Pass:**

Durch diese Regeln erfolgt die Umsetzung der URLs auf die Struktur des lokalen Dateisystems. Zumindest die folgende Regel sollte hier stehen:

```
Pass /* <document_root>/*
```

Für <document\_root> ist das Verzeichnis anzugeben, in dem die HomePage (Begrüßungs-Seite) des Servers steht (auch DocumentRoot genannt). Stehen z.B. alle Ihre

HTML-Dokumente einschl. der HomePage unter /local/WWW, bekommt ein Client auf seine Anforderung mit dem URL `http://<servername>/inside/servers.html` das Dokument /local/WWW/inside/servers.html geschickt. Es sind mehrere Pass-Regeln möglich.

### Erster Test

Diese Konfiguration genügt, um einen ersten Versuch zu wagen (reicht aber nicht aus, um damit auf Dauer einen Server zu betreiben!). Kopieren Sie eine einfache HTML-Seite mit dem Namen welcome.html in Ihr Document-Root. Starten Sie den httpd von der Kommandozeile:

```
httpd &
```

und überprüfen Sie sein Vorhandensein in der Prozeßtafel. Versuchen Sie nun, mit einem Browser Ihr Dokument vom Server anzufordern:

```
http://<servername>/welcome.html
```

Als Alternative dazu besteht auch die Möglichkeit, den Server über Telnet zu testen:

```
telnet <servername> <port>,
```

also z.B.

```
telnet www.hu-berlin.de 80
```

Ein Login-Prompt erscheint nicht. Sie müssen eingeben:

```
GET /welcome.html
```

Entweder bekommen Sie das Dokument im ASCII-Format (nebst einigen HTTP-Protokoll-Meldungen) oder eine Fehlermeldung angezeigt.

### Fehlersuche

Falls beim Test ein Fehler aufgetreten ist, beenden Sie den Server-Prozess (kill ..) und starten ihn anschließend im Verbose Mode: `httpd -v &`.

Weitere Hinweise finden sich evtl. in den unter Access-Log, ErrorLog benannten Log-Dateien. Dort kann jedoch nur etwas stehen, wenn der Server-Prozess (UserId nobody o.ä.!) Schreibrechte in diesem Verzeichnis hat. Es ist deshalb günstig, ihm dafür ein Verzeichnis zuzuteilen, in dem er kaum Schaden anrichten kann, z.B. /tmp oder ein Verzeichnis logs unter ServerRoot, das z.B. nobody gehört.

Im DocumentRoot (und weiteren Unterverzeichnissen) muß die Welt Leserechte haben (außer natürlich bei Dokumenten, die Sie nur einem ausgewählten Personenkreis zugänglich machen wollen)!

### Verbesserung der Installation

Wenn dieser Test erfolgreich war, ist es Zeit, sich um die dauerhafte Installation des Servers zu kümmern. Zunächst sollte entschieden werden, ob der Server-Prozess beim Hochfahren des Rechners automatisch gestartet („Stand-alone Installation“) oder nur auf Anforderungen hin vom inet-Dämon aktiviert wird.

#### a) Server-Start beim Hochfahren

Es ist eine Eintragung in einer System-Startup-Datei notwendig (Root-Rechte erforderlich!). Der Name dieser Datei ist systemabhängig (Beispiele: System V.4 (z.B.

Sun Solaris 2.x): /etc/init.d mit einem Link nach /etc/rc3.d, Linux: /etc/rc.d/rc.inet2 oder /etc/rc.d/rc.local). In der Dokumentation (→ 2.4.1 Stand-alone Installation) ist ein Beispiel zu finden.

b) Server-Start unter inetd

In /etc/services muß folgende Zeile stehen:

```
http 80/tcp #WWW-Server
```

(Falls Ihr Server einen anderen Port als den Standard-Port bedienen soll, muß statt 80 dessen Nummer hier stehen. Der Port darf in dieser Datei nicht schon vergeben sein. Bei Maschinen, auf denen NIS (Yellow Pages) läuft, ist diese Eintragung dort vorzunehmen.).

In /etc/inetd.conf ist einzutragen:

```
http stream tcp nowait root <serverroot>/httpd httpd
```

(<serverroot> entspricht Ihrer Eintragung unter Server-Root in der Konfigurationsdatei. Hinter dem letzten httpd müssen evtl. erforderliche Parameter folgen.

Nach dieser Änderung muß inetd zum Restart aufgefordert (kill -HUP <inetd\_prozessnummer>) bzw. die Maschine neu gestartet werden!

Danach sollte erneut getestet werden (→ erster Test), ob der Server auf Anforderungen richtig reagiert.

### Weitere Hinweise zur Konfigurationsdatei:

Wenn Ihr Server-Prozeß ständig läuft (also nicht jedesmal vom inetd „geweckt“ wird), müssen sie nach jeder Änderung seiner Konfigurationsdateien einen Restart veranlassen! Am einfachsten geht das mit: `httpd -restart`.

- **HostName** <servername inklusive domainname>

Um Schwierigkeiten bei der Namensauflösung zu umgehen, sollte hier der volle Hostname angegeben werden. Falls der Server einen Aliasnamen (→ Teil2, Planung vor dem Start) hat, z.B. www, sollte dieser hier stehen.

Beispiel: HostName www.hu-berlin.de

- **PidFile** <datei>

In diese Datei trägt der Server die Prozeßnummer ein, unter der er läuft. Das Vorhandensein dieser Zeile gewährleistet, daß beim Restart des Servers mit http -restart die Datei mit der Prozeßnummer zuverlässig gefunden wird. Die Angabe kann mit vollem Pfadnamen oder relativ zum ServerRoot erfolgen.

- **Welcome** <datei>

Beispiel: Welcome index.html

Bei fehlender Dateiangabe im URL (URL wird mit einem Schrägstrich abgeschlossen) wird im angeforderten Verzeichnis nach einer Datei index.html gesucht und diese dann angezeigt. Mehrere solcher Regeln sind möglich.

- **UserDir** <verzeichnis>

Legt der Benutzer in seinem HOME-Verzeichnis ein Directory mit dem Namen <verzeichnis> an und platziert in dieses seine HTML-Dokumente, kann über einen URL der Form:

```
http://<servername>/~<nutzerkennzeichen>/...
```

auf diese Dokumente zugegriffen werden.

Winfried Naumann

### Quellen:

- [1] World-Wide Web Server Software: <http://www.w3.org/hypertext/WWW/Daemon/Overview.html>  
HTTP: Servers: [http://www.yahoo.com/Computers/World\\_Wide\\_Web/HTTP/Servers](http://www.yahoo.com/Computers/World_Wide_Web/HTTP/Servers)
- [2] Thomas Boutell: World Wide Web FAQ, [http://sunsite.unc.edu/boutell/faq/www\\_faq.html](http://sunsite.unc.edu/boutell/faq/www_faq.html)
- [3] Cricket Liu, Jerry Peek, Russ Jones et.al.: Managing Internet Information Services. World Wide Web, Gopher, FTP, and more., O'Reilly & Associates, Inc., Sebastopol, 1994
- [4] J. Seeger: Jenseits von HTML, Dritte Internationale 3W-Konferenz in Darmstadt, in: iX 6/1995, S. 8  
V. Weber: Surfer's Days, Dritte WWW-Konferenz, in: c't 6/1995, S. 32  
Third International World-Wide-Web Conference: <http://www.igd.fhg.de/www95.html>  
The International WWW Conference Series: <http://www.igd.fhg.de/www/www95/conference.html>
- [5] W.Naumann: WWW, in: RZ-Mitteilungen Nr.8, S.23 ff.
- [6] Übersicht HTML-Editoren: HTML Editors, <http://mirror.wwa.com/spike/95/htmls.html>  
HTML Editors, <http://union.ncsa.uiuc.edu/HyperNews/get/www/html/editors.html>

### CERN httpd:

- [7] CERN HTTP server status: <http://www.w3.org/hypertext/WWW/Daemon/Status.html>
- [8] R. Klute: Serviertechnik. Installation und Konfiguration des CERN-WWW-Servers, in: iX, 3/1994, S.182 ff.
- [9] Platforms supported bei Library, HTTP Server and Line Mode Browser: <http://www.w3.org/hypertext/User/Platform/>
- [10] Dokumentation: (siehe auch <ftp://ftp.rz.hu-berlin.de/>)  
HTML, online: CERN httpd 3.0 Guide: <http://www.w3.org/hypertext/WWW/Daemon/User/Guide.html>  
HTML, tar-Archiv: Postscript: <http://www.w3.org/hypertext/WWW/Daemon/User/Guide.ps>
- [11] Binaries: <http://www.w3.org/pub/bin/> (siehe auch <ftp://ftp.rz.hu-berlin.de/>)

### NCSA httpd:

- [12] NCSA httpd: <http://hoohoo.ncsa.uiuc.edu/docs/Overview.html>
- [13] Plattformen (Auswahl): Precompiled Executables: <http://hoohoo.ncsa.uiuc.edu/docs/setup/PreCompiled.html>
- [14] Dokumentation: (siehe auch <ftp://ftp.rz.hu-berlin.de/>)  
HTML, online: NCSA httpd, <http://hoohoo.ncsa.uiuc.edu/docs/Overview.html>  
HTML, tar-Archiv: [http://ftp.ncsa.uiuc.edu/Web/httpd/Unix/ncsa\\_httpd/documents/httpd-1.4docs.html](http://ftp.ncsa.uiuc.edu/Web/httpd/Unix/ncsa_httpd/documents/httpd-1.4docs.html)  
Postscript: <http://hoohoo.ncsa.uiuc.edu/docs/postscript-docs/Overview.html>
- [15] Binaries: [ftp://Web/httpd/UNIX/ncsa\\_httpd/](ftp://Web/httpd/UNIX/ncsa_httpd/) (siehe auch <ftp://ftp.rz.hu-berlin.de/>)