

Proseminar:
Website-Management-Systeme

Thema:
WebServer: Apache/Roxen

von Oliver Roeschke

eMail: o_roesch@informatik.uni-kl.de

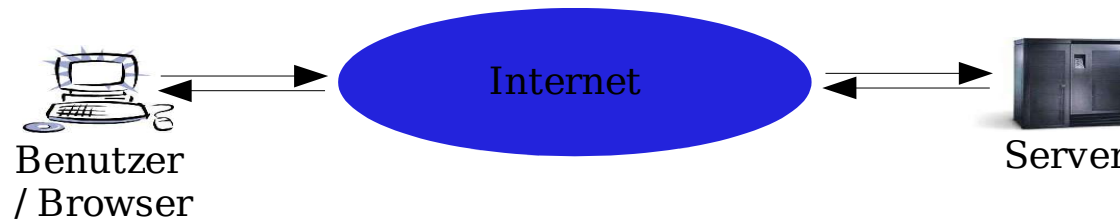
Gliederung:

- 1.) kurze Einleitung
- 2.) Begriffsklärung
- 3.) Was ist ein WebServer?
- 4.) das HTTP-Protokoll
- 5.) Verarbeitung in Apache
- 6.) HTML
- 7.) Quellenverzeichnis

1.) Kurze Einleitung:

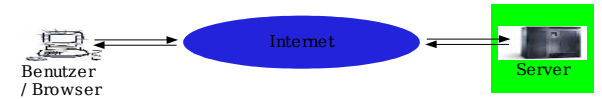
Dieser Vortrag behandelt die grundlegende Kommunikation zwischen dem Benutzer/ Browser und dem WebServer, sowie dem dazu verwendeten Protokoll (HTTP).

Außerdem wird die interne Verarbeitung in Apache schematisch dargestellt und die Zusammenhänge von HTML mit Content-Management gezeigt.



2.) Begriffsklärung:

| | |
|------------|--|
| Request | meint die Anfrage des Benutzers/ Browsers an den WebServer |
| Host | Name unter dem ein Server verfügbar ist |
| TCP/IP | T ransmission C ontrol P rotocol/ I nternet P rotocol – Protokolle die Verbindungen zwischen Knoten des Internets erlauben. |
| IP-Adresse | eindeutige numerische Adresse, mit deren Hilfe TCP/IP die Kommunikation zwischen Endknoten koordiniert |
| DNS | D omain N ame S ystem – System das Domain- und Hostnamen in IP-Adressen auflöst |



3.) Was ist ein WebServer?

Definition:

Ein Webserver ist eine Software die es ermöglicht Dokumente bereitzustellen und unter zu Hilfenahme von HTTP zu transferieren.

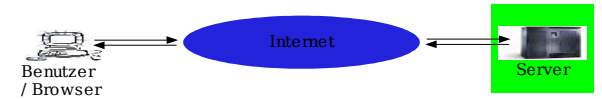
Beispiele für Dokumente:

Internetseiten, Downloads, eShops, Online-Anmeldesysteme

Der Webserver muss unter einer eindeutigen Adresse bekannt sein. Diese besteht aus:

<Maschine>.<Domain>.<TLD>

| | | |
|-------|-------------------------|--|
| wobei | <Maschine> | der Name des Rechners (z.B. 'caladrius') |
| | <Domain> | die Domain (z.B. 'homelinux') |
| | <TLD> | die Top-Level-Domain (z.B. 'net') |

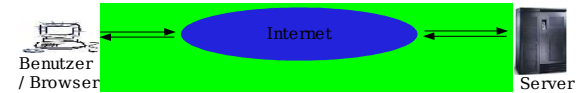


3.) Was ist ein WebServer? [2]

Der Versand der Daten, erfolgt nur auf Anfrage ('Request') durch einen Benutzer/ Browser.

Dokumente die abgefragt werden, müssen nicht im HTML-Format dem Web-Server vorliegen. Das Ausführen von Skripten (PHP, CGI, Perl, usw.) ist in Apache möglich.

Solche Skripte erzeugen als Ausgabe HTML-Code der dem Benutzer angezeigt werden kann.



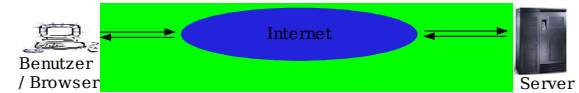
4.) HTTP-Protokoll

kurzer geschichtlicher Überblick:

| | | |
|------|----------|--|
| 1990 | HTTP/1.0 | erste HTTP Implementierung für den Transport von Dokumenten (RFC 1945) |
| 1993 | HTTP/1.0 | erste Veröffentlichung im freien Browser Mosaic |
| 1997 | HTTP/1.1 | Einführung von praxisorientierten Erweiterungen, heute meist verwendeter Standard (RFC 2616) |

wichtigsten Unterschiede HTTP/1.0 ↔ HTTP/1.1:

- Mehrfachverwendung einer HTTP-Verbindung
- Angabe des Hostnamens bei der Anfrage



4.) HTTP-Protokoll [2]

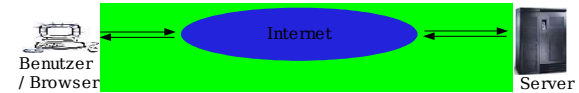
Die Anfrage:

Beim Aufruf einer Webseite (z.B. 'http://caladrius.homelinux.net') durch den Benutzer stellt der Browser (z.B. Konqueror) eine Anfrage an den WebServer. Hier findet das HTTP-Protokoll seine Verwendung.

Hierzu wird zunächst der Hostname per DNS in eine IP-Adresse aufgelöst (hier: '127.0.0.1').

Danach baut der Browser eine TCP/IP-Verbindung zu der ermittelten IP-Adresse an Port 80 auf.

Ist dies gelungen, so steht eine bi-direktionale TCP/IP Verbindung zwischen dem WebServer und dem Browser, welche als Kommunikationsmedium für das HTTP-Protokoll dient.



4.) HTTP-Protokoll [3]

Daten die per HTTP transportiert werden sind nicht verschlüsselt und werden in Klartext codiert.

Ein Request muss folgend formatiert sein:

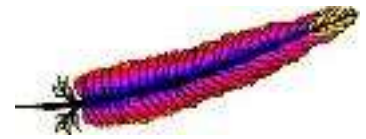
```
GET <STRING> HTTP/1.1  
HOST: <HOST>
```

wobei **<STRING>** Zeichenkette die durch den WebServer ausgewertet wird

<HOST> Name des Hostes (z.B. 'caladrius.homelinux.net')

In unserem Beispiel also:

```
GET / HTTP/1.1  
Host: caladrius.homelinux.net
```



5.) Verarbeitung in Apache

Zuerst wird bestimmt ob es sich bei **<STRING>** um ein HTML-Dokument oder ein Skript handelt, welches ausgeführt werden muss.

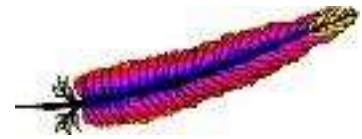
Ist **<STRING>** eine Datei wird versucht das nötige Dokument im Datenspeicher aufzufinden.

Wurde nur ein Verzeichnis übergeben (z.B. '/') ergänzt Apache dies automatisch, Voreinstellung ist 'index.html'.

- existiert Dokument nicht, wird Fehler '404 Not found' zurückgegeben

Nun werden die Sicherheitseinstellungen für das Dokument geprüft, um sicherzustellen das kein Benutzer unberechtigten Zugang zu dem Dokument erhält.

Hat der Benutzer keine Zugriffsrechte auf die Datei wird Fehler '401 Unauthorized' zurückgegeben.

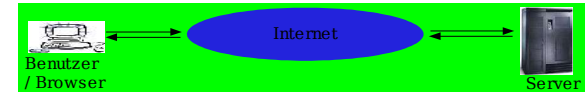


5.) Verarbeitung in Apache [2]

Kann Apache das Dokument ermitteln und hat der Benutzer Zugriff darauf, gibt der WebServer die Meldung 'HTTP/1.1 200 OK' mit Informationen über den Inhalt der transferierten Datei, gefolgt von dem Inhalt der Datei in Klartext. In unserem Beispiel sieht die angeforderte Datei so aus:

```
<HTML> <HEAD>
<title>Test-Seite 1</title> </HEAD>
<BODY> dies ist eine einfach test-seite fuer das proseminar bei
der ag softwaretechnik <br>
<a href="/test/test.html">Test-Link</a><br>
</BODY></HTML>
```

Auf diese Weise können auch andere Dokumente (Bilder, Downloads, usw.) per HTTP übertragen werden.



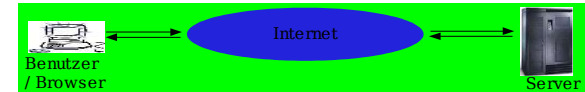
6.) HTML

Gestaltung und Aufbau von Webseiten wird mit Hilfe der Auszeichnungssprache HTML (**H**yper**T**ext **M**arkup **L**anguage) realisiert.

Verknüpfungen ('Hyperlinks') in HTML-Dokumenten zu anderen Dokumenten ermöglichen die Verknüpfung von Inhalten (daher 'HyperText').

Verwendung von TAGs um Formatierung des Textes festzulegen ('Markup'). Dabei haben HTML-TAGs immer die gleiche definierte Funktion, lassen daher nur begrenzten Formatierungs-Spielraum.

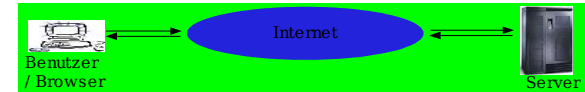
Mit HTML ist keine Trennung von Layout und Inhalt eines Dokumentes möglich, eignet sich daher lediglich zum Transfer von Dokumenten zu Betrachter (Benutzer/ Browser), nicht aber zur effizienten Verwaltung der Inhalte.



6.) HTML [2]

wichtige HTML-TAG's:

- <HTML> </HTML>** markieren Anfang und Ende des HTML-Dokuments
- <HEAD> </HEAD>** markieren Anfang und Ende des Kopfes, welcher Metadaten über das darzustellende Dokument enthalten kann
- <BODY> </BODY>** markieren Anfang und Ende des sichtbaren Dokumentbereiches
-
TEXT ** erzeugt Hyperlink zu LINK auf TEXT (auch andere Objekte können Hyperlinks präsentieren)
- ** fügt BILD in Dokument ein



6.) HTML [3]

Viele Formatierungseigenschaften werden durch Eigenschaften innerhalb der TAGs festgelegt, z.B. Farbeinstellungen für Texte, Links, usw.

Um die Funktionsbeschränktheit von HTML zu umgehen wird heute in aufwendigeren Webseiten regelmäßig JavaScript, Flash, ActiveX und weitere Techniken verwendet.



7.) Literaturverzeichnis

- ❑ HTML 4.01 Spezifikation
<http://www.w3c.org/TR/html4/>
- ❑ RFC 2616 - HTTP/1.1
<http://www.w3c.org/Protocols/rfc2616/rfc2616.html>
- ❑ RFC 1945 - HTTP/1.0
<http://www.w3.org/Protocols/rfc1945/rfc1945>
- ❑ Apache Documentation
<http://httpd.apache.org/docs-2.0>