



# 7363 - Web-basierte Anwendungen: **Übung 03**

Installation und Konfiguration eines  
Web-Servers am Beispiel Apache 2.0



## Ziele



- Fähigkeit, einen eigenen Web-Server für die Projektarbeiten im Rahmen dieses Kurses einzurichten
  - Keine spezielle Sicherheitsschulung, da nur lokaler Betrieb
  - Keine Hochleistungskonfiguration (hier nicht benötigt)
  - Keine Berücksichtigung von Effizienzfragen bei der Administration (wir wollen hier keine Serverkonsolidierung betreiben)
- Damit bleiben:
  - Installation
    - incl. Kompilierung, im Rahmen der automatisierten Abläufe
  - Konfiguration
    - Grundkenntnisse
  - Protokolle eines Web-Servers
    - Grundkenntnisse
  - Ergänzung eigener Module (!)
    - Hinreichende Kenntnisse zum Einrichten der für den Kurs benötigten Module



# Installation



## Installation



- Vorbemerkung
  - Dies ist eine Einzelübung - bitte eine Installation pro Person!
  - Erfahrungsaustausch innerhalb der Teams ist ok.
- Quellen
  - Material in `~werntges/lv/wba/03`, sofern nicht anders erwähnt. Dies sei der Wert von `$srcdir`.
  - Dokumentation: Entweder in den Paketen oder im Internet unter URLs, die z.B. in den Paketen erwähnt sind.
- Der Server
  - Apache ("a patchy server") in der aktuellen Version 2.0.53
  - Datei: `httpd-2.0.53.tar.bz2`
- Vorbereitungen: **Schaffen Sie Platz!**
  - Löschen unbenutzter Dateien
  - Leeren Ihrer Browser-Caches
  - Archivierung + Kompression noch benötigter Daten



## 1. Auspacken:

```
$ cd # ggf. anpassen
$ tar tvjf $srcdir/httpd-2.53.tar.bz2 # Inhalte listen
# Weiter, wenn ok; sonst Verzeichnis wechseln, dann:
$ tar xjf $srcdir/httpd-2.53.tar.gz
$ cd httpd-2.0.53
```

## 2. Doku lesen, insbesondere:

README, INSTALL, Internet-Seiten zu "Install"  
Rufen Sie die Optionen von "configure" ab

## 3. PREFIX festlegen

- Dies ist der zukünftige Installationsort
- Legen Sie eine rein lokale Version an, d.h. unter Ihrem Verzeichnis und mit Ihren Berechtigungen.
- Vorschlag: `$HOME/apache2`



## 4. Probeinstallation

- Die folgenden Schritte sind bei Bedarf zu wiederholen, ggf. mit verschiedenen Optionen von `configure` und/oder angepassten Umgebungsvariablen

```
$ ./configure --prefix=$HOME/apache2 # bzw. Ihre Wahl
$ make
$ make install
```

## 5. Bemerkungen

- Im Idealfall ist der Server nun bereits startklar. Im Allgemeinen muss er aber noch konfiguriert werden, was mehr Arbeit ist als die Installation.
- Zur Installation neuer Module ist manchmal eine komplette Neuinstallation notwendig. Mit `$ make distclean` können Sie die ausgepackten Verzeichnisse in den Anfangszustand zurücksetzen.



# Konfiguration



## Konfiguration



- Wechseln Sie zu `$HOME/apache2`
  - Im Unterverzeichnis `conf` befinden sich verschiedene Versionen einer Konfigurationsdatei.
  - Die Datei "httpd.conf" ist die tatsächlich wirksame. Sie ist anfangs identisch mit "httpd-std.conf"
- Editieren Sie **httpd.conf**
  - Machen Sie sich vertraut mit den Inhalten dieser Datei,
  - diskutieren Sie die Einträge mit Ihrem Projektpartner.
  - Vorgabe: Der Server soll auf "localhost", Port 8888 laufen.
    - Ein ungenutzter Port mit hoher Nummer steht Ihnen auch ohne Adminrechte zur Verfügung. Port 80 ist auch schon belegt.
    - "localhost" ist ausreichend (und sicher) für isolierte Tests. Für teamübergreifende Tests ist statt dessen der Name Ihres Servers zu verwenden, z.B. "lx2-03".
  - Ändern Sie den Eintrag in "Listen" entsprechend.



- Bemerkungen
  - Es gibt zahlreiche Gründe für Änderungen dieser Konfigurationsdatei
  - Die Administration eines Web-Servers besteht zu erheblichen Teilen aus der Optimierung dieser Datei (zumindest in der Aufbauphase).
  - Sie werden daher gelegentlich zu dieser Datei zurückkehren.



## Erste Tests



## Erste Tests



- Starten Sie den Server probeweise

```
$ cd ~/apache2
$ ./bin/apachectl start
# Fehlermeldungen? Kontrolle der Prozesse:
$ ps -ef # Laufen httpd-Prozesse unter Ihrem Account?
$ ./bin/apachectl stop
```
- Tipp:
  - Verwenden Sie "apachectl restart" nach Änderungen in httpd.conf
- Erste statische Dokumente abrufen
  - Server starten
  - Rufen Sie den URL "http://localhost:8888/" bzw. "/" ab. Verwenden Sie dazu anfangs "telnet" wie in der Vorlesungs-Demo, dann Ihren Browser.
  - Rufen Sie dann einen nicht vorhandenen URL ab, etwa "http://localhost:8888/nosuchfile".
  - Erklären Sie, woher die Antworten stammen.
    - Das ist bei Apache komplizierter als zunächst vermutet!
  - Ändern Sie diese testweise (zum Beweis)!



## Erste Tests



- Erste virtuelle Dokumente abrufen
  - Fordern Sie "/" erneut an, diesmal aber **auf deutsch** (wenn Sie es vorhin auf englisch erhielten, sonst umgekehrt)
  - Nutzen Sie dazu HTTP-Header zur Angabe der von Ihnen bevorzugten Sprache!
  - Was macht Ihr Server dabei?
    - Stichwort in der Apache-Doku: „Content negotiation“
  - Finden Sie Hinweise auf den Mechanismus der Dokumentenzuordnung in httpd.conf?



- Die Protokolldateien:
  - logs/access.log
  - logs/error.log
  - httpd.pid
- Sichten Sie die Inhalte
  - Erklären Sie deren Zustandekommen.
- Tipp:
  - Verwenden Sie zum Debugging immer diese Dateien.



- Dynamisch erzeugte Seiten
  - Greifen Sie auf `http://localhost:8888/cgi-bin/test-cgi` zu.
  - Der Zugriff scheitert. Finden Sie die Ursache und beheben Sie sie.
  - Was erhalten Sie als Anzeige?
  - Verfahren Sie analog mit `http://localhost:8888/cgi-bin/printenv`
- Variationen in der Konfiguration
  - Aktivieren Sie das "Referer-Log". Testen Sie seine Wirkung.
  - Ändern Sie das "KeepAliveTimeout" von 15 auf 30 Sekunden
  - Ergänzen Sie MIME-Types zu Dateiendungen
    - \*.text (analog zu \*.txt)
    - \*.ent (external parsed entities, etwa die Dateien aus dem Unterbau-Projekt)
    - Machen Sie sich dazu mit der Zuordnung zwischen Dateiendungen und MIME Types (im HTTP-Header) vertraut.



## Einbau von Zusatzmodulen

... am Beispiel FastCGI  
(nächste Übung; in dieser Woche  
als freiwilliges Extra für schnelle  
Gruppen, zumindest zum Einlesen)



## Zusatzmodul FastCGI



- Quelle
  - `$srcdir/mod_fastcgi-2.4.2.tar.gz`.
- Installation
  - Quellpaket entpacken, in das entstehende Unterverzeichnis `mod_fastcgi-2.4.2.tar.gz` wechseln.
  - Anweisungen in `INSTALL.AP2` des Pakets befolgen
    - Makefile anpassen (Installationsverzeichnis / „top\_dir“!)
- Konfiguration
  - Legen Sie ein Verzeichnis "fcgi-bin" an, analog zu "cgi-bin" (gleiches Elternverzeichnis, gleiche Rechte)
  - In `httpd.conf`:
    - Eintrag „LoadModule ...“ gemäß Anleitung einfügen
    - Einträge „ScriptAlias“ und „Directory“ für fcgi-bin anlegen, Fälle für „cgi-bin“ als Vorbilder nehmen.
  - Neues Verzeichnis mit "fastcgi-script" assoziieren:
    - `<Location /fcgi-bin> SetHandler fastcgi-script </Location>`



- Installation von FCGI-Anwendungsprogrammen
  - FCGI-Skript/-Programm in „fcgi-bin“ kopieren
  - Execute- und Leserechte!
  - Programm bei Apache anmelden – Eintrag in httpd.conf:
    - `AppClass path-to-my_fcgi_module.fcgi`
  
- Wirkung:
  - Bereits beim Start des Servers kann Apache die angemeldeten FCGI-Module als Prozesse starten
  - Bei Aufruf per URL sind die Prozesse bereits in Betrieb und initialisiert, können also direkt mit der Aufgabe beginnen.
  - Beachte: Es gibt mehrere FCGI-Betriebsarten:
    - Statische Anwendungen (Direktive „FastCgiServer“)
    - Dynamische Anwendungen (Direktive „FastCgiConfig“)
    - Externe Anwendungen (Direktive „FastCgiExternalServer“)
  
- Dazu später mehr ...