



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

Installation und Konfiguration  
eines Web-Servers

Apache 2.x, Lighttpd 1.4.11



- 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



- Teilziele
  - Fertigstellung einer Entwicklungsumgebung für dynamische Web-Seiten.
  - Vertraut werden mit dem CGI bzw. seiner Alternative
- Übungen
  - Performance-Vergleich: CGI vs. Alternativen (FastCGI)
  - Umgang mit Formularen
  - Umgang mit SSI und Cookies
- Übergeordnetes Ziel
  - Einübung von Grundlagen-Fertigkeiten für den späteren Einsatz von Web Services
  - Vorbereitungen für die Projektarbeit



# Installation

Apache  
„Lighty“



- 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.
- Apache
  - Apache ("a patchy server") in der getesteten Version 2.0.53 oder in der aktuellen Version 2.2.3
  - Dateien: `httpd-2.0.53.tar.bz2` oder `httpd-2.2.3.tar.bz2`
- „Lighty“
  - Datei: `lighttpd-1.4.11.tar.gz`
- Vorbereitungen: **Schaffen Sie Platz!**
  - Löschen unbenutzter Dateien
  - Leeren Ihrer Browser-Caches
  - Archivierung + Kompression noch benötigter Daten

**ACHTUNG Lighty-Nutzer:**  
Lighty ist inzwischen vorinstalliert,  
das Binary befindet sich dort:  
`/usr/sbin/lighttpd`



## 1. Auspacken (Bsp.):

```
$ 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`



## 1. Auspacken (Bsp.):

```
$ cd                # ggf. anpassen
$ tar tvxf $srcdir/lighttpd-1.4.11.tar.gz # Inhalte listen
# Weiter, wenn ok; sonst Verzeichnis wechseln, dann:
$ tar xzf $srcdir/lighttpd-1.4.11.tar.gz
$ cd lighttpd-1.4.11
```

## 2. Doku lesen, insbesondere:

README, INSTALL, Internet-Seiten (<http://www.lighttpd.net>)

## 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/lighttpd`



## 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.



## 4. Probeinstallation

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

```
$ ./configure --prefix=$HOME/lighttpd # bzw. Ihre Wahl
```

```
$ make
```

```
$ make install
```

**ACHTUNG Lighty-Nutzer:**  
Kompilieren ist optional,  
z.B. bei Problemen mit dem  
vorinstallierten „Lighty“

## 5. PATH-Anpassung

```
$ export PATH=$PATH:~/lighttpd/bin:~/lighttpd/sbin
```

```
# Besser: USERPATH in ~/.bashrc analog anpassen!
```

## 6. 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



- 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.



- Legen Sie ein Arbeitsverzeichnis für den Betrieb Ihrer Lighty-Installation an, z.B. `$HOME/var/lighttpd`
  - Legen Sie dort Unterverzeichnisse `conf` und `logs` an.
  - „conf“ nimmt die Konfigurationsdatei auf, „logs“ die Protokolldateien für Fehler und allgemeine Zugriffe sowie für die Process ID.
- **lighttpd.conf**
  - Folgen Sie der Anleitung: Legen Sie in „conf“ eine Kopie der Konf.-Datei aus dem Installationsbereich an und passen Sie diese an.
    - Machen Sie sich vertraut mit den Inhalten dieser Datei.
    - Diskutieren Sie die Einträge mit Ihrem Projektpartner.
    - Beschäftigen Sie sich mit den Beispielen und Erläuterungen auf der Projektseite von Lighttpd.
  - Ziel: Umsetzung der folgenden Vorgaben, Testbetrieb



- **lighttpd.conf: Vorgaben**

- 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".
- Logs, PID-Datei
  - Verwenden Sie die Standardnamen für diese drei Einträge
  - Konfigurieren Sie als Verzeichnis das soeben angelegte „logs“ (s.o.)
- Module
  - Aktivieren Sie mindestens folgende Module:  
`mod_rewrite, mod_redirect, mod_access, mod_fastcgi,`  
`mod_cgi, mod_compress, mod_ssi, mod_accesslog`
- Server-Root
  - Legen Sie das Bezugs-Verzeichnis für HTTP-Request fest
  - Beispiel: `$HOME/public_html`



- Bemerkungen
  - Es gibt zahlreiche Gründe für Änderungen der 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



- Apache

```
$ 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

- Lighty

```
cd ~/var/lighttpd/conf
```

```
$ lighttpd -t -f lighttpd.conf # Syntax prüfen
```

```
# Ggf. Fehler in conf-Daten beseitigen, dann:
```

```
$ lighttpd -f lighttpd.conf
```

```
# Fehlermeldungen? Kontrolle der Prozesse:
```

```
$ ps -ef # Laufen httpd-Prozesse unter Ihrem Account?
```

```
$ kill `cat ../logs/lighttpd.pid` # Prozess beenden
```



# Erste Tests (Apache)

---



- 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)!
- Lighty
  - Sollten Sie noch keine HTML-Datei zum Testen zur Verfügung haben, richten Sie eine in Ihrem „server-root“-Verzeichnis ein.
  - Verwenden Sie z.B. die HTML-Ergebnisse aus der DocBook-Übung.



# Erste Tests (nur Apache)

---



- 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 bzw. logs/lighttpd.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`
  - Nur Lighty: Richten Sie vorher cgi-bin ein, kopieren Sie test-cgi und printenv aus \$srcdir dorthin
- Variationen in der Konfiguration
  - Aktivieren Sie das "Referer-Log". Testen Sie seine Wirkung.
  - Ändern Sie das "KeepAliveTimeout" z.B. von 15 auf 30 Sekunden
  - Ergänzen Sie MIME-Types zu Dateiendungen
    - \*.text (analog zu \*.txt), \*.xhtml (welcher Typ ist hier der richtige??)
    - \*.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



- 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 ...



- Installation
  - Dieses Modul ist bereits Teil von Lighttpd und muss nicht separat installiert werden.
- Konfiguration
  - Legen Sie ein Verzeichnis "fcgi-bin" an, analog zu "cgi-bin" (gleiches Elternverzeichnis, gleiche Rechte)
  - In lighttpd.conf:
    - Eintrag „fastcgi.server“ ent-kommentieren und ausfüllen
    - Für jedes zu startende FCGI-Script einen Eintrag wie folgt einrichten:

```
"anmelden.fcgi" =>  
( "localhost" =>  
  (  
    "socket" => "/tmp/anmelden.socket",  
    "bin-path" => "/pfad/zu/ihrem/anmelden.fcgi"  
  )  
)
```



# Zusatzmodul FastCGI (Lighty)

---



- Installation von FCGI-Anwendungsprogrammen
  - FCGI-Skript/-Programm in „fcgi-bin“ kopieren
  - Execute- und Leserechte!
  - Programm bei Lighttpd anmelden – Eintrag in lighttpd.conf, s.o.
- Wirkung:
  - Bereits beim Start des Servers kann Lighty die angemeldeten FCGI-Module als Prozesse starten (auch mehrere zur Lastverteilung!)
  - Bei Aufruf per URL sind die Prozesse bereits in Betrieb und initialisiert, können also direkt mit der Aufgabe beginnen.



# Einbau weiterer Module



- Perl bzw. Ruby, CGI / FCGI
  - Ihr Web-Server enthält nun die erforderlichen Module
  - Verzichten Sie auf mod\_perl bzw. mod\_ruby, falls Sie FCGI einsetzen
- PHP-Anwender
  - Ergänzen Sie selbständig Ihre Apache-Installation um die für PHP notwendigen Module, testen Sie diese.
- PHP: Teilnehmer-Tipps
  - Apache 2 mit PHP5 als FastCGI:  
[http://www.howto24.de/wiki/index.php/Apache\\_2\\_mit\\_PHP5\\_als\\_FastCGI#Installation\\_von\\_mod\\_fastcgi](http://www.howto24.de/wiki/index.php/Apache_2_mit_PHP5_als_FastCGI#Installation_von_mod_fastcgi)
  - Sicherer Webserver mit PHP und FastCGI:  
<http://www.commy.de/2005/08/06/howto-sicherer-webserver-mit-php-fastcgi/>
  - PHP-Handbuch, Kap. 42: Safe mode  
<http://de.php.net/features.safe-mode>



- Java-Anwender
  - Installieren Sie Apache Tomcat bzw. den für Ihr Projekt benötigten Web-Server
  - Entscheiden Sie sich für die richtige Version
  
- Andere, sofern vorhanden
  - Ergänzen Sie Ihre Apache-Installation um die benötigten Module, bis dass die von Ihnen gewählte Entwicklungsumgebung für dynamische Web-Seiten testbereit ist.