

# Web-Server: Minimalistischer Apache

Wer nur statische HTML-Seiten oder Grafiken anbieten möchte, kann auf viele Funktionen vom Apache verzichten. Über ein zusätzliches Apache-Modul sind jedoch auch dynamische Seiten mit Perl möglich.

Ziel dieser Übung ist, die Größe des Apache-Verzeichnisses auf unter 400 kB zu reduzieren.

Von der Durchführung dieser Konfiguration auf produktiven Systemen rate ich ausdrücklich ab.

Als Basis brauchen wir natürlich einige Devel-Pakete sowie Compiler und Tools:

gcc, make, lininclude & glibc-devel

Als Quellcode-Paket nehmen wir die aktuelle, stabile Apache-Version von [httpd.apache.org](http://httpd.apache.org) (oder einem Spiegelserver). Das Archiv sollte `apache_[VERSION].tar.gz` heißen. Entpacken mit:

```
tar xfvz apache_[VERSION].tar.gz
```

```
cd apache[VERSION]
```

Der nächste Schritt ist das Configure-Script mit den entsprechenden Parametern zu starten. Hier kommt die Feinarbeit: alle standardmäßig eingebundenen Module werden deaktiviert; die Module 'dir' und 'mime' mit dem Apache kompiliert (nicht als Modul).

```
./configure
  --prefix=/usr/local/apache
  --without-support
  --enable-module=dir
  --disable-shared=dir
  --enable-module=mime
  --disable-shared=mime
  --disable-module=cgi
  --disable-module=auth
  --disable-module=log_config
  --disable-module=usertrack
  --disable-module=env
  --disable-module=negotiation
  --disable-module=status
  --disable-module=include
  --disable-module=autoindex
  --disable-module=asis
  --disable-module=imap
  --disable-module=actions
```

# Web-Server: Minimalistischer Apache

```
--disable-module=userdir  
--disable-module=alias  
--disable-module=access  
--disable-module=setenvif
```

Optional können auch folgende Module noch mit einkompiliert werden:

```
--enable-module=log_config  
--disable-shared=log_config
```

Zum Protokollieren von Zugriffen und Fehlern braucht man Log-Modul.

```
--enable-module=auth  
--disable-shared=auth
```

Das Auth-Modul braucht man um Bereiche der Website nur für bestimmte Benutzer zugänglich zu machen

```
--enable-module=cgi  
--disable-shared=cgi
```

Mit dem CGI-Modul können CGI-Programme (Perl-/PHP-Skripte) auf dem Server ausgeführt werden.

Wurde das Configure-Script ohne Fehlermeldung beendet, starten wir den compiler:

```
make
```

Ist auch dieser Vorgang ohne Fehler beendet worden, kann der Apache installiert werden:

```
make install
```

Um das Apache-Verzeichnis von überflüssigem Ballast zu befreien, können folgende Verzeichnisse und Dateien gelöscht werden:

```
cd /usr/local/apache/  
rm -rf cgi-bin/  
rm -rf icons/  
rm -rf man/  
rm -rf proxy/  
rm -rf htdocs/manual/
```

# Web-Server: Minimalistischer Apache

```
rm -rf libexec/  
rm -rf include/  
rm -f conf/*.default  
rm -f conf/srm.conf  
rm -f conf/access.conf  
rm -f htdocs/*
```

Nun kommt die Apache-Konfigurationsdatei (`/usr/local/apache/conf/httpd.conf`) an die Reihe:

```
ServerName router.linux-fuer-alle.de  
ServerAdmin root  
ServerType standalone  
ServerRoot "/usr/local/apache"  
PidFile /usr/local/apache/logs/httpd.pid  
Timeout 300  
KeepAlive On  
MaxKeepAliveRequests 100  
KeepAliveTimeout 15  
MinSpareServers 1  
MaxSpareServers 10  
StartServers 1  
MaxClients 10  
MaxRequestsPerChild 0  
Port 80  
Listen 192.168.1.1:80  
User nobody  
Group nobody  
  
DocumentRoot "/usr/local/apache/htdocs"  
DefaultType text/plain  
  
# folgende Zeile für mod_mime aktivieren  
TypesConfig conf/mime.types  
  
# folgende Zeile für mod_dir aktivieren  
DirectoryIndex index.html  
  
# folgende Zeile für mod_cgi aktivieren  
# AddHandler cgi-script .pl  
  
# nächste Zeile für mod_auth aktivieren  
# AccessFileName .htaccess  
  
# für CGI-Programme/Perl-Scripte " ExecCGI" an folgende Zeile anhängen  
Options Indexes
```

# Web-Server: Minimalistischer Apache

```
HostnameLookups Off
ErrorLog /usr/local/apache/logs/error_log
LogLevel warn
```

Vor dem ersten Start, sollte noch ein Inhalt im htdocs-Verzeichnis (`/usr/local/apache/htdocs/`) angelegt werden:

```
<HTML>
<HEAD>
<TITLE>Apache-Test</TITLE>
</HEAD>
<BODY>
<P>Apache-Test</P>
</BODY>
```

Nun kann der Apache gestartet werden:

```
/usr/local/apache/bin/httpd
```

Prüfen ob er läuft:

```
ps ax | grep httpd
```

In einem Internet-Browser die URL angeben:

```
http://192.168.1.1/
```

Den Webserver schließlich wieder beendet:

```
killall httpd
```

Schlußbemerkung

Alternativ kann man sich auch mal das Paket [micro\\_httpd](#) ansehen.

*Eindeutige ID: #1040*

*huschi*

*2005-12-11 22:14*