



1.104.7 Systemdateien finden und Dateien am richtigen Ort platzieren



www.lpi.org



Copyright (©) 2006 by [W. Kicherer](#). This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

FHS (Filesystem Hierarchie Standard)

/	Wurzelverzeichnis
/bin	Ausführbare Programme (binaries), welche für den grundlegenden Systembetrieb wichtig sind
/sbin	Ausführbare Programme, welche für den Super-User (root) vorgesehen und den grundlegenden Systembetrieb wichtig sind
/etc	Konfigurationsdateien des Systems
/lib	Bibliothek (Shared Libraries), Programmteile und Routinen, welche von mehreren Programmen benutzt werden
/dev	Geräte-Dateien (Festplatten, Drucker, Modems, ...)
=> Alle auf der Root-Partition!!!!!!!!!!	
/boot	Programme und Dateien, welche zum Booten verwendet werden (z.B. Kernel-Binary, Initial Ramdisk, Bootloader)
/home	Home-Verzeichnis aller User (Ausnahme: root)
/tmp	Temporary Files, d.h. Dateien die nur kurz verwendet werden und dann wieder gelöscht werden können
/var	Dateien, welche sich ständig ändern: Log-Dateien, Spool-Verzeichnisse, ...
/usr	Dateien welche sich selten ändern und für das grundlegende System nicht nötig sind



/usr	
/usr/X11R6	Das X Window System, Version 11, Release 6
/usr/bin	Die meisten User Kommandos, welche nicht grundlegend sind
/usr/include	Die Include-Dateien des C-Compilers (Header)
/usr/lib	Bibliotheken für Programme und Pakete
/usr/local	Die dritte, lokale Hierarchie
/usr/sbin	Nicht lebensnotwendige Systemprogramme
/usr/share	Geteilte Daten für alle Architekturen
/usr/src	Der Quellcode für Programme
/var	
/var/lib	Libraries, welche nicht grundlegend sind
/var/lock	Lock-Dateien, um etwas zu sperren
/var/log	Log-Dateien, zur Systemkontrolle
/var/run	Prozessinfos zu gerade laufenden Programmen
/var/spool	Warteschlangenverzeichnis, z. B. Für den Drucker, für Mails
/var/tmp	Temporäre Dateien



- **Spezielle Verzeichnisse müssen in die Root-Partition**
- **Grober Überblick über die Verzeichnishierarchie**

which: durchsucht die Verzeichnisse, welche durch die Umgebungsvariable \$PATH vorgegeben sind nach Programmen

```
root@abc:/# which mount  
/bin/mount
```

whereis: Wie which, findet aber auch Handbuchseiten und Sourcen

```
root@abc:/# whereis mount  
mount: /bin/mount /sbin/mount.smb  
/sbin/mount.smbfs /sbin/mount.captive /usr/  
share/man/man2/mount.2.gz  
/usr/share/man/man8/mount.8.gz
```

type: Wie which, gibt aber auch Shell-interne Befehle und Aliase an

```
root@abc:/# type ls  
ls is aliased to `ls --color=auto'
```



find: Durchsucht ganze Verzeichnishierarchien nach Dateinamen (Muster wie in der Shell), nach Änderungsdatum, Dateirechten, Dateigröße, ... Und kann auch Aktionen auf den gefundenen Dateien ausführen lassen

Syntax:

find *Startverzeichnis* *Test* *Aktion*

Beispiel:

find **/bin** **-name "[Mm][ae][iy]er"** **-print**

(-print kann auch weggelassen werden, weitere Aktionen möglich; hier Muster wie in der Shell; auch -regex)

```
root@abc:~# find /usr/X11R6/ -name "[Xx]*config*"
/usr/X11R6/bin/xf86config-v3
/usr/X11R6/bin/xf86config
/usr/X11R6/include/X11/extensions/XKBconfig.h
/usr/X11R6/man/man1/xf86config-v3.1x.gz
/usr/X11R6/man/man1/xf86config.1x.gz
/usr/X11R6/man/man3/XReconfigureWMWindow.3x.gz
```



- **find** durchsucht direkt die Verzeichnisse und ist deswegen langsam
- Schneller wäre es, wenn nur eine Datenbank durchsucht werden muss
- **locate** → mit **updatedb** wird eine Datenbank angelegt, welche dann von **locate** durchsucht wird
- **locate** durchsucht die von **updatedb** angelegte Datenbank nach Dateinamen (Muster wie in der Shell)
- Beispiel:

```
~root@abc:~# locate "xf86config"
/usr/sbin/mkxf86config
/usr/X11R6/bin/xf86config
/usr/X11R6/bin/xf86config-v3
```
- **updatedb** läuft i.d.R einmal am Tag via **cron**
- Konfiguration via **/etc/updatedb.conf** – dort u.a. Angaben welche Verzeichnisse/Dateisysteme nicht durchsucht werden sollen.
- **slocate** (secure locate) verhält sich erst einmal wie **locate**. Es werden aber auch die Zugriffsrechte berücksichtigt und nur Dateien gelistet auf die der User auch Rechte hat.



- **Wo liegt das Programm mount?**
- **Wo liegt das Programm cd?**
- **Wo liegt die Man-Page zu cat?**
- **Suchen Sie alle Dateien mit dem Namen „fstab“.**
- **Suchen Sie alle Dateien, die jünger als 2 Tage sind.**



- **which, whereis, type** suchen nur in **\$PATH** und nur Programme bzw. inkl. Man-Pages und Sources
- **find** sucht im angegebenen Pfad, nach Namen, Datum, Größe, Rechten, ... und kann auch Aktionen ausführen
- **locate (slocate)** sucht nach Dateinamen in einer Datenbank. Ergebnis ist evtl. veraltet (letzter Durchgang von updatedb) aber schneller als find.