

SelfLinux-0.12.3



Verzeichnisse unter Linux



Autor: Frank Boerner (frank@frank-boerner.de)
Formatierung: Matthias Hagedorn (matthias.hagedorn@selflinux.org)
Lizenz: GPL

Die Linux-Verzeichnisstruktur und das Arbeiten mit Verzeichnissen

Inhaltsverzeichnis

1 Linux-Verzeichnisstruktur

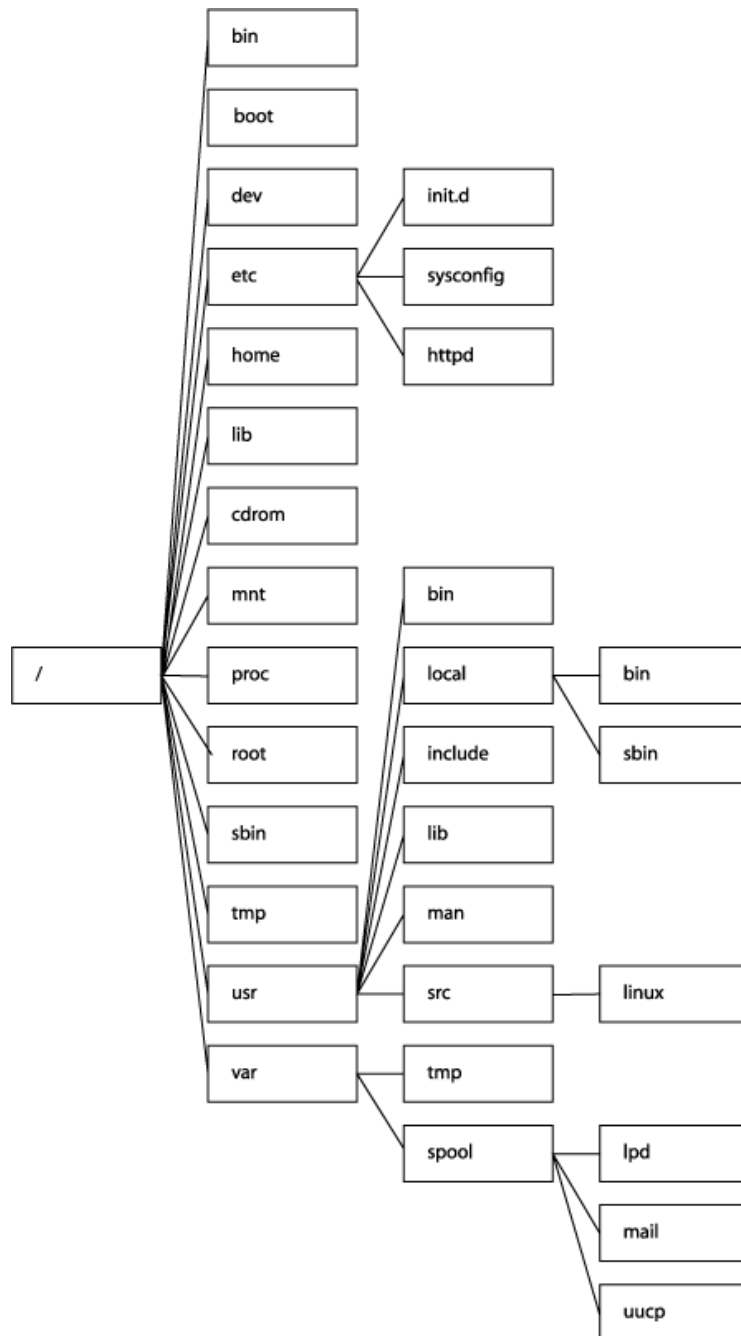
2 Arbeiten mit Verzeichnissen

- 2.1 das Kommando pwd
- 2.2 das Kommando cd
- 2.3 das Kommando ls
- 2.4 das Kommando mkdir
- 2.5 das Kommando rmdir

1 Linux-Verzeichnisstruktur

Linux speichert Dateien **im Gegensatz zu MS-DOS und Windows** in einem großen Verzeichnisbaum, der mit dem Wurzelverzeichnis / beginnt und sich schnell weit verzweigt.

Die nachfolgende Grafik stellt nur einen Ausschnitt eines Systems dar und muss nicht in allen Einzelheiten auf jedes System zutreffen.



Ausschnitt eines Systems

Die wichtigsten Verzeichnisse sind:

- / Das Wurzelverzeichnis steht ganz oben in der Hierarchie.
- /bin Hier befinden sich wichtige Programme für Anwender, die immer verfügbar sein müssen, z. B. die

	Shells.
<code>/boot</code>	Hier befinden sich die zum Hochfahren des Systems unbedingt erforderlichen Dateien. In der Hauptsache ist das der Kernel, im Normalfall eine Datei mit dem Namen <code>vmlinuz</code> . Aber auch andere Namen sind möglich.
<code>/dev</code>	Dieses Verzeichnis enthält nur Spezialdateien, sogenannte Gerätedateien. Diese stellen eine einfach zu nutzende Schnittstelle zur Hardware dar. Hier finden sich auch Einträge für alle Festplatten und ihre Partitionen: <code>/dev/hda</code> ist die erste EIDE-, <code>/dev/sda</code> die erste SCSI-Festplatte im System. Höhere Buchstaben (<code>hdb</code> , <code>hdc</code>) stellen weitere Festplatten dar, Zahlen am Ende (<code>sda1</code> , <code>sda2</code>) sind die Partitionen der Festplatten. Da auf einer Festplatte nur vier primäre Partitionen möglich sind, wird häufig eine erweiterte Partition angelegt, die den größten Teil der Festplatte umfasst. In der erweiterten Partition können dann "logische Laufwerke" angelegt werden. Diese erhalten grundsätzlich die Partitionsnummern ab 5. Enthält eine Festplatte also eine primäre und eine erweiterte Partition, in der sich wiederum zwei logische Laufwerke befinden, gibt es auf dieser Platte die Partitionen 1, 2, 5 und 6. Die primäre Partition ist 1, die erweiterte ist 2, und die beiden logischen Laufwerke sind 5 und 6.
<code>/etc</code>	Hier sind viele der Konfigurationsdateien untergebracht, welche die Einstellungen verschiedener Programme oder auch grundlegende Systeminformationen enthalten. Viele dieser Dateien haben ihre eigene Manpage in Kapitel 5.
<code>/home</code>	In diesem Verzeichnis liegen traditionell die Heimatverzeichnisse der Benutzer des Systems.
<code>/lib</code>	Hier befinden sich die wichtigsten Funktionsbibliotheken des Systems. Eigentlich gibt es nur eine Grundregel im Umgang mit diesem Verzeichnis: Finger weg!
<code>/proc</code>	<code>/proc</code> ist eigentlich kein normales Verzeichnis, sondern stellt eine Schnittstelle zum Kernel dar. Jedes laufende Programm wird hier in einem Unterverzeichnis geführt, dessen Dateien viele Informationen z.B. über den aktuellen Programmstatus enthalten. Zudem gibt es eine umfangreiche Verzeichnisstruktur mit Daten über den Kernel und die Hardware des Systems.
<code>/root</code>	Dies ist das Heimatverzeichnis des Systemverwalters <code>root</code> . Es liegt traditionell im Wurzelverzeichnis, damit <code>root</code> auch dann auf seine Dateien (beispielsweise Diagnoseprogramme) zugreifen kann, wenn durch einen Fehler der Zugriff auf andere Partitionen nicht mehr möglich ist.
<code>/sbin</code>	Ähnlich wie <code>/bin</code> enthält auch <code>/sbin</code> wichtige Programme. Diese sind jedoch hauptsächlich für den Systemverwalter gedacht, da sie Funktionen erfüllen, auf die ein normaler Benutzer keinen Zugriff hat.
<code>/tmp</code>	Dieses Verzeichnis kann von jedem Benutzer und jedem Programm als temporäre Ablage für Dateien verwendet werden. Damit sich Benutzer nicht gegenseitig ihre Dateien löschen, ist das sogenannte Sticky-Bit dieses Verzeichnisses gesetzt.
<code>/usr</code>	Die umfangreichste Verzeichnisstruktur des Systems. Hier liegt der größte Teil der installierten Software. Auf vielen Systemen befinden sich in und unterhalb von <code>/usr</code> mehr Daten als in allen anderen Dateien zusammen. Die Programmdateien sind meist in <code>/usr/bin</code> , die Spiele in <code>/usr/games</code> . In Netzwerken, an die viele gleichartige Systeme angeschlossen sind, wird dieses Verzeichnis häufig auf einem zentralen Server gespeichert, und alle anderen Computer greifen über das Netzwerk darauf zu.
<code>/var</code>	Unter diesem Verzeichnis werden hauptsächlich Dateien gespeichert, die sich ständig verändern. Der Name <code>/var</code> steht für variabel , also veränderlich . Hier befinden sich beispielsweise die Verzeichnisse für ausgehende E-Mail und noch ungelesene eingehende, wenn der jeweilige Benutzer nicht dafür gesorgt hat, dass neue E-Mails automatisch in sein Heimatverzeichnis übertragen werden.
<code>/opt</code>	(optionale Software) Kommerzielle Software oder sehr große Programme, die nicht unmittelbar zum System gehören, wie etwa <i>KDE</i> , <i>Netscape</i> , <i>Mozilla</i> usw. finden hier ihren Platz.

Wenn Sie ein ext2- oder ext3-Dateisystem verwenden, kann es vorkommen, dass ein Verzeichnis `/lost+found` in Ihrem Wurzelverzeichnis existiert. In dieses Verzeichnis werden Dateien abgelegt, die keinem anderen Verzeichnis mehr zugeordnet werden können. Solche Dateien, die in sehr seltenen Fällen z.B. bei

Programmabstürzen oder Hardwarefehlern entstehen können, werden von einem Prüfwerkzeug des Dateisystems gefunden und dann in `/lost+found` abgelegt. Es handelt sich hier um eine besondere Eigenschaft der ext-Dateisysteme.

(Die Originalbeschreibung der Verzeichnisinhalte nach dem *Filesystem Hierarchy Standard* - *FHS* ist nachzulesen unter [🇬🇧 http://www.pathname.com/fhs/](http://www.pathname.com/fhs/)

2 Arbeiten mit Verzeichnissen

2.1 das Kommando pwd

Das Kommando gibt den vollständigen Pfadnamen des im Moment aktuellen Arbeitsverzeichnisses an.

```
user@linux ~/cvs/tutorial/userbasics/konzepte_basic/ $ pwd
/home/user/cvs/tutorial/userbasics/konzepte_basic
```

2.2 das Kommando cd

Ohne Aufruf von Argumenten wechselt das Kommando von jedem beliebigen Punkt des Dateisystems in das Homeverzeichnis des Benutzers.

```
user@linux ~/cvs/tutorial/userbasics/konzepte_basic/ $ cd
user@linux ~/ $
```

Wenn man ein Verzeichnis wechseln möchte, unterscheidet man zwischen relativer und absoluter Pfadangabe.

Beispiel relativer Pfad:

```
user@linux ~/ $ cd cvs/tutorial
user@linux ~/cvs/tutorial/ $
```

Hier wechselt der Benutzer ausgehend von seinem Heimatverzeichnis in das Verzeichnis `cvs/tutorial`.

Beispiel absoluter Pfad:

```
user@linux ~/ $ cd /home/linux/cvs/tutorial
user@linux ~/cvs/tutorial/ $
```

Bei absoluter Pfadangabe wird vom Wurzelverzeichnis an jedes einzelne Verzeichnis angegeben.

Mit dem Kommando `cd ..` wechseln Sie immer in das Elternverzeichnis Ihres aktuellen Verzeichnisses. Befinden Sie sich bereits im Wurzelverzeichnis, bleibt dieses Kommando wirkungslos.

```
user@linux ~/cvs/tutorial/ $ cd ..
user@linux ~/cvs/ $
```

2.3 das Kommando ls

Das Kommando `ls` ist das wahrscheinlich wichtigste und am häufigsten aufgerufene Kommando unter Linux überhaupt: Es zeigt den Inhalt eines Verzeichnisses an und ist damit die Entsprechung von DIR unter DOS.

Ruft man `ls` ohne Optionen auf, so gibt es den Inhalt in der kürzesten möglichen Form aus:

```

user@linux ~/ $ ls
Desktop      OpenOffice.org641  playlists  public_html      wrg
Documents    cvs                 nsmail     plugin131.trace  selflinux
Mail         evolution          office52   plugin131_02.trace
selflinux-texte
user@linux ~/ $

```

Die verschiedenen Optionen zu `ls` erlauben die Anzeige einer Vielzahl von Information in verschiedenen Formaten. Wir möchten hier nur auf die wichtigsten Optionen eingehen. Für weitere Möglichkeiten geben Sie einfach `ls --help` ein.

```
ls -a
```

Diese Option listet auch die sogenannten versteckten Dateien auf, die mit einem Punkt beginnen.

```

user@linux ~/ $ ls -a
.                .fullcircle     .nautilus
.xsession        .gconf          .nautilus-metafile.xml
..               .gconfd         .netscape
.xsession-errors .DCOPserver_linux_0 .gconfd         .netscape6      .y2log
.xtalkrc         .DCOPserver_linux__0 .gimp-1.2      .ntrc            .yast2
.G-Force         .gnome          .opera
.ICEauthority    .gnome-desktop  .pinerc
Desktop          .SuSENautilusPage .gnome-help-browser
Documents        .gnome-private  .profile
.Xauthority      .gnome_private  .profile          Mail

```

```
ls -l
```

Diese Option liefert eine ausführlichere Ausgabe. Sie bedeutet von links nach rechts: Dateityp, Zugriffsrechte, Linkzähler, Besitzer, Gruppe, Größe, Datum der letzten Änderung, Dateiname

```

user@linux ~/ $ ls -l
insgesamt 18
drwxr-xr-x  3 user  users  520 Mai 28 20:10 Desktop
drwxr-xr-x  3 user  users 1072 Mai 28 21:41 Documents
drwx----- 29 user  users 4800 Mai 28 21:59 Mail
drwxr-xr-x  3 user  users  320 Apr 13 00:58 OpenOffice.org641
drwxr-xr-x  7 user  users  168 Mai 26 21:47 cvs
drwx-----  4 user  users  232 Apr 14 08:04 evolution
drwxr-xr-x  2 root  root   208 Feb 24 2000 kuer_chm
drwx-----  2 user  users   48 Apr 13 01:09 nsmail
drwxr-xr-x  3 user  users  232 Apr 14 08:07 office52
drwxr-xr-x  2 user  users   80 Mai  4 07:39 playlists
drwxr-xr-x  2 user  users   80 Apr 11 20:04 public_html
drwxr-xr-x  5 user  users  208 Mai  8 20:42 selflinux
drwxr-xr-x  2 user  users  352 Mär 16 13:48 selflinux-texte

```



```
drwxr-xr-x  2 user  users          448 Mär 16 13:47 wrg
```

`ls -R`

Diese Option gibt auch den Inhalt der Unterverzeichnisse mit aus.

```
user@linux ~/selflinux/ $ ls -R
.:
Basisrelease README output saxon slcompile styles
./output:
html images
./output/html:
Basisrelease.html Basisrelease01.html
./output/images:
10baset-netz.gif img_transparent.gif openfolder.gif
shell_funktionsweise01.gif
fddi.gif kdm.gif repeater_bridge.gif
thickethernet.gif
img_document.gif koaxialnetz.gif selflinux.gif tokenring.gif
./saxon:
README doc saxon.jar
./saxon/doc:
api-guide.html conformance.html extensions.html patterns.html
xsl-elements.html
changes.html dtdgen.html history.html samples.html
changes5.html expressions.html index.html suggestions.html
conditions.html extensibility.html instant.html using-xsl.html
./styles:
html
./styles/html:
files.xsl format.xsl lib.xsl main.xsl section.xsl selflinux.xsd
toc.xsl
```

Sie können auch mehrere Optionen miteinander verbinden, zum Beispiel `ls -lR` oder `ls -la`.

2.4 das Kommando `mkdir`

Mit dem Kommando `mkdir` kann man Verzeichnisse anlegen.

```
user@linux ~/test/ $ ls
user@linux ~/test/ $ mkdir a
user@linux ~/test/ $ ls

a
```

Durch Verwendung der Option `-p` kann gleichzeitig auch noch Unterverzeichnisse anlegen.

```
user@linux ~/test/ $ ls
user@linux ~/test/ $ mkdir -p a/b/c
user@linux ~/test/ $ ls

a

user@linux ~/test/ $ cd a
user@linux ~/test/a/ $ ls

b

user@linux ~/test/a/ $ cd b
user@linux ~/test/a/b/ $ ls

c

user@linux ~/test/a/b/ $
```

2.5 das Kommando `rmdir`

Mit dem Kommando `rmdir` kann man (leere) Verzeichnisse löschen.

```
user@linux ~/test/ $ ls

a

user@linux ~/test/ $ rmdir a
user@linux ~/test/ $ ls
user@linux ~/test/ $
```

Wenn Sie hier die Option `-p` verwenden, werden gleichzeitig auch die untergeordneten Verzeichnisse gelöscht (vorausgesetzt, dass alle leer sind).

```
user@linux ~/test/ $ ls

a

user@linux ~/test/ $ rmdir -p a/b/c
user@linux ~/test/ $ ls
user@linux ~/test/ $
```