

2. Dateisystem, Pfade und Verzeichnisse

- [2.1 Dateisystem, Pfade und Verzeichnisse](#)

2.1 Dateisystem, Pfade und Verzeichnisse

Linux verwendet einen hierarchischen Verzeichnisbaum.

Das bedeutet:

Alle Dateien,
Verzeichnisse,
Geräte,
Mountpoints
und Systembereiche befinden sich unter einem gemeinsamen Startpunkt.

Dieser Startpunkt heißt:

/

Das ist das Root-Verzeichnis des Dateisystems.

Wichtig:

Das Root-Verzeichnis / ist nicht dasselbe wie der Benutzer root.

Merksatz:

Unter Linux beginnt alles bei /.

Lernziele

Nach dieser Seite solltest du erklären können:

- wie der Linux-Verzeichnisbaum aufgebaut ist
- was absolute und relative Pfade sind
- was das Root-Verzeichnis ist
- was das Home-Verzeichnis ist
- welche wichtigen Systemverzeichnisse es gibt
- was versteckte Dateien sind
- was Mountpoints sind

- warum unter Linux Laufwerke nicht wie C: oder D: eingebunden werden

Grundidee des Linux-Dateisystems

Unter Linux gibt es keinen Laufwerksbuchstaben wie bei Windows.

Windows-Beispiel:

```
C:\Users\Felix\Dokumente
```

Linux-Beispiel:

```
/home/felix/Dokumente
```

Unter Linux werden Datenträger, Partitionen, USB-Sticks, Netzlaufwerke oder andere Dateisysteme in den bestehenden Verzeichnisbaum eingehängt.

Dieses Einhängen nennt man:

```
Mounten
```

Merksatz:

```
Linux nutzt einen gemeinsamen Verzeichnisbaum,  
keine Laufwerksbuchstaben.
```

Root-Verzeichnis /

Das Root-Verzeichnis ist der oberste Punkt im Linux-Dateisystem.

Es wird dargestellt durch:

```
/
```

Alle anderen Verzeichnisse liegen darunter.

Beispiele:

/home

/etc

/var

/usr

/tmp

/root

/dev

/proc

Wichtig:

/ ist der Anfang des Dateisystems.

root ist der administrative Benutzer.

Merksatz:

/ ist ein Verzeichnis.

root ist ein Benutzer.

Verzeichnisbaum

Ein einfacher Linux-Verzeichnisbaum kann so aussehen:

```
/
├── bin
├── boot
├── dev
├── etc
├── home
│   └── felix
├── root
```

```
├─ tmp
├─ usr
└─ var
```

Bedeutung:

/ ist der oberste Punkt.

/home/felix ist ein Benutzerverzeichnis.

/etc enthält viele Konfigurationen.

/var enthält veränderliche Daten wie Logs.

Merksatz:

Pfade beschreiben die Position im Verzeichnisbaum.

Absolute Pfade

Ein absoluter Pfad beginnt immer bei:

```
/
```

Beispiele:

```
/home/felix
```

```
/etc/ssh/sshd_config
```

```
/var/log
```

```
/usr/bin
```

```
/tmp/test.txt
```

Vorteil:

Ein absoluter Pfad ist eindeutig,
egal in welchem Verzeichnis man sich gerade befindet.

Merksatz:

Absoluter Pfad beginnt mit /.

Relative Pfade

Ein relativer Pfad beginnt nicht bei /, sondern bezieht sich auf das aktuelle Verzeichnis.

Beispiel:

dokumente/datei.txt

Wenn man sich gerade in

/home/felix

befindet, dann bedeutet der relative Pfad:

/home/felix/dokumente/datei.txt

Merksatz:

Relativer Pfad hängt vom aktuellen Verzeichnis ab.

Aktuelles Verzeichnis

Das aktuelle Verzeichnis ist der Ort, an dem man sich gerade in der Shell befindet.

Anzeigen mit:

pwd

pwd steht für:

```
print working directory
```

Beispielausgabe:

```
/home/felix
```

Merksatz:

```
pwd zeigt,  
wo du gerade bist.
```

In Verzeichnisse wechseln

Mit `cd` wechselt man das aktuelle Verzeichnis.

Beispiele:

```
cd /etc  
  
cd /home/felix  
  
cd ..  
  
cd ~  
  
cd -
```

Bedeutung:

Befehl	Bedeutung
<code>cd /etc</code>	wechselt nach /etc
<code>cd ..</code>	eine Ebene nach oben
<code>cd ~</code>	ins eigene Home-Verzeichnis
<code>cd -</code>	zurück ins vorherige Verzeichnis
<code>cd</code>	ebenfalls meist ins eigene Home-Verzeichnis

Merksatz:

```
cd wechselt das Verzeichnis.
```

Punkt und Doppelpunkt

Unter Linux haben `.` und `..` eine besondere Bedeutung.

Zeichen	Bedeutung
<code>.</code>	aktuelles Verzeichnis
<code>..</code>	übergeordnetes Verzeichnis

Beispiele:

```
./script.sh
```

bedeutet:

```
script.sh im aktuellen Verzeichnis ausführen
```

```
cd ..
```

bedeutet:

```
eine Ebene nach oben wechseln
```

Merksatz:

```
. ist hier.
```

```
.. ist eine Ebene höher.
```

Home-Verzeichnis

Jeder normale Benutzer hat meist ein eigenes Home-Verzeichnis.

Beispiel:

```
/home/felix
```


Dort liegen persönliche Dateien, Konfigurationen und Benutzerdaten.

Typische Inhalte:

Dokumente

Downloads

Skripte

versteckte Konfigurationsdateien

persönliche Einstellungen

Das Home-Verzeichnis kann mit ~ abgekürzt werden.

Beispiel:

~/Dokumente

bedeutet:

/home/felix/Dokumente

Merksatz:

~ steht für das eigene Home-Verzeichnis.

Root-Benutzer und /root

Der administrative Benutzer heißt:

root

Sein Home-Verzeichnis ist meist:

/root

Das ist nicht dasselbe wie:

```
/
```

Unterschied:

Begriff	Bedeutung
/	Root-Verzeichnis des gesamten Dateisystems
root	administrativer Benutzer
/root	Home-Verzeichnis des Benutzers root

Merksatz:

```
/,  
root  
und /root sind drei verschiedene Dinge.
```

Wichtige Linux-Verzeichnisse

Verzeichnis	Bedeutung
/	Wurzel des Dateisystems
/home	Home-Verzeichnisse normaler Benutzer
/root	Home-Verzeichnis des Root-Benutzers
/etc	Konfigurationsdateien
/var	veränderliche Daten
/var/log	Logdateien
/tmp	temporäre Dateien
/usr	Programme, Bibliotheken und Ressourcen
/bin	wichtige Benutzerprogramme
/sbin	wichtige Systemprogramme
/boot	Bootdateien und Kernel
/dev	Gerätedateien
/proc	Prozess- und Kernelinformationen
/sys	Kernel- und Geräteinformationen
/mnt	temporäre Mountpoints

Verzeichnis	Bedeutung
/media	automatisch eingebundene Wechselmedien
/opt	optionale Zusatzsoftware
/srv	Daten für bereitgestellte Dienste
/run	Laufzeitdaten seit Systemstart

Merksatz:

/etc für Konfiguration.

/var/log für Logs.

/home für Benutzer.

/etc

Das Verzeichnis /etc enthält viele systemweite Konfigurationsdateien.

Beispiele:

/etc/ssh/sshd_config

/etc/fstab

/etc/hosts

/etc/passwd

/etc/group

/etc/resolv.conf

Typische Inhalte:

Dienstkonfiguration

Netzwerkkonfiguration

Benutzer- und Gruppendateien

Mount-Konfiguration

Systemkonfiguration

Merksatz:

/etc ist zentral für Konfigurationen.

/var

Das Verzeichnis /var enthält veränderliche Daten.

var steht für:

variable

Typische Inhalte:

Logs

Spool-Dateien

Caches

Daten von Diensten

Warteschlangen

Beispiele:

/var/log

/var/cache

/var/spool

/var/lib

Merksatz:

/var enthält Daten,
die sich im Betrieb ändern.

/var/log

Das Verzeichnis /var/log enthält viele Logdateien.

Logs helfen bei:

Fehlersuche

Dienstproblemen

Loginproblemen

Sicherheitsanalyse

Systemereignissen

Beispiele:

/var/log/syslog

/var/log/auth.log

/var/log/messages

/var/log/nginx

/var/log/apache2

Wichtig:

Welche Logdateien vorhanden sind,
hängt von Distribution und Diensten ab.

Merksatz:

/var/log ist wichtig für Fehlersuche.

/tmp

Das Verzeichnis /tmp enthält temporäre Dateien.

Typische Nutzung:

Zwischenspeicher

temporäre Skriptdateien

Installationsreste

kurzfristige Arbeitsdateien

Wichtig:

Daten in /tmp können automatisch gelöscht werden,
zum Beispiel beim Neustart oder durch Aufräumdienste.

Merksatz:

/tmp ist nicht für dauerhafte Speicherung gedacht.

/usr

Das Verzeichnis /usr enthält viele Programme, Bibliotheken und gemeinsam genutzte Ressourcen.

Typische Unterverzeichnisse:

/usr/bin

/usr/sbin

/usr/lib

/usr/share

Beispiele:

Programme für Benutzer

Systemprogramme

Bibliotheken

Dokumentation

Merksatz:

/usr enthält viele installierte Programme und Ressourcen.

/bin und /sbin

/bin enthält wichtige Benutzerprogramme.

Beispiele:

ls

cp

mv

rm

cat

/sbin enthält wichtige Systemprogramme.

Beispiele:

ip

reboot

shutdown

fsck

Hinweis:

Auf modernen Distributionen sind `/bin` und `/sbin` teilweise symbolische Links nach `/usr/bin` und `/usr/sbin`.

Merksatz:

`bin` = Programme.

`sbin` = Systemprogramme.

/boot

Das Verzeichnis `/boot` enthält Dateien, die für den Systemstart wichtig sind.

Typische Inhalte:

Kernel

Bootloader-Dateien

Initramfs

Wichtig:

Fehler in `/boot` können dazu führen,
dass das System nicht mehr startet.

Merksatz:

`/boot` enthält wichtige Startdateien.

/dev

Das Verzeichnis `/dev` enthält Gerätedateien.

Unter Linux werden viele Geräte als Dateien dargestellt.

Beispiele:

`/dev/sda`

`/dev/nvme0n1`

`/dev/null`

`/dev/tty`

`/dev/random`

Bedeutung:

Festplatten

Partitionen

Terminals

virtuelle Geräte

Merksatz:

Unter Linux erscheinen Geräte oft als Dateien unter `/dev`.

`/proc`

Das Verzeichnis `/proc` ist ein virtuelles Dateisystem.

Es enthält Informationen über:

Prozesse

Kernel

Speicher

CPU

Systemzustand

Beispiele:

/proc/cpuinfo

/proc/meminfo

/proc/1

Wichtig:

/proc liegt nicht einfach normal auf der Festplatte.

Es wird vom Kernel bereitgestellt.

Merksatz:

/proc zeigt laufende Kernel- und Prozessinformationen.

/sys

Das Verzeichnis /sys ist ebenfalls ein virtuelles Dateisystem.

Es enthält Informationen über:

Geräte

Treiber

Kernelobjekte

Hardwarezustände

Wichtig:

/sys wird häufig von Systemdiensten,
Hardwareverwaltung
und Diagnosewerkzeugen genutzt.

Merksatz:

/sys zeigt System- und Geräteinformationen.

/mnt und /media

/mnt wird häufig für temporäre manuelle Mountpoints genutzt.

Beispiel:

/mnt/backup

/mnt/usb

/media wird häufig für automatisch eingebundene Wechselmedien genutzt.

Beispiel:

/media/felix/USB-STICK

Merksatz:

/mnt für manuelle Mounts.

/media oft für Wechselmedien.

/opt

Das Verzeichnis /opt ist für optionale Zusatzsoftware gedacht.

Beispiele:

herstellerspezifische Programme

manuell installierte Anwendungen

große Zusatzpakete

Merksatz:

/opt enthält optionale Zusatzsoftware.

/srv

Das Verzeichnis /srv kann Daten enthalten, die von Serverdiensten bereitgestellt werden.

Beispiele:

Webdaten

FTP-Daten

Projektdaten für Dienste

Merksatz:

/srv kann Dienst-Daten enthalten.

/run

Das Verzeichnis /run enthält Laufzeitdaten seit dem Systemstart.

Beispiele:

PID-Dateien

Sockets

temporäre Dienstinformationen

Wichtig:

Inhalte in /run sind nicht dauerhaft.

Merksatz:

/run enthält aktuelle Laufzeitinformationen.

Versteckte Dateien

Dateien und Verzeichnisse, deren Name mit einem Punkt beginnt, gelten unter Linux als versteckt.

Beispiele:

.bashrc

.profile

.ssh

.config

.local

Anzeigen mit:

ls -a

Typische Nutzung:

Benutzerkonfigurationen

SSH-Schlüssel

Programmeinstellungen

Merksatz:

Punkt am Anfang bedeutet versteckt.

Dateitypen unter Linux

Unter Linux gibt es verschiedene Dateitypen.

Typ	Bedeutung
normale Datei	Textdatei, Programm, Bild, Archiv
Verzeichnis	enthält Dateien und weitere Verzeichnisse
symbolischer Link	Verweis auf andere Datei oder Verzeichnis
Gerätedatei	Zugriff auf Geräte
Socket	Kommunikation zwischen Prozessen
Pipe	Datenstrom zwischen Prozessen

Mit folgendem Befehl kann man Dateiinformationen anzeigen:

```
ls -l
```

Merksatz:

```
Unter Linux ist nicht jede „Datei“ eine normale Datei.
```

Symbolische Links

Ein symbolischer Link ist ein Verweis auf eine andere Datei oder ein anderes Verzeichnis.

Beispiel:

```
/bin kann auf /usr/bin zeigen
```

Symbolische Links erkennt man bei `ls -l` häufig an:

```
->
```

Beispiel:

```
bin -> usr/bin
```

Nutzen:

Kompatibilität

Abkürzungen

flexible Pfade

Verweise auf gemeinsame Orte

Merksatz:

Symbolischer Link = Verweis.

Mountpoints

Ein Mountpoint ist ein Verzeichnis, an dem ein Dateisystem eingehängt wird.

Beispiel:

/mnt/backup

Dort kann zum Beispiel eine externe Festplatte, eine Partition oder ein Netzlaufwerk eingebunden werden.

Wichtig:

Das Dateisystem erscheint dann an dieser Stelle im Verzeichnisbaum.

Merksatz:

Mountpoint = Einhängpunkt im Verzeichnisbaum.

Warum Linux keine Laufwerksbuchstaben nutzt

Windows nutzt oft Laufwerksbuchstaben:

C:

D:

E:

Linux nutzt stattdessen einen einheitlichen Verzeichnisbaum.

Beispiel:

/

/home

/mnt/backup

/media/usb

Vorteil:

Alle Dateisysteme erscheinen an einer Stelle im Baum.

Programme können mit festen Pfaden arbeiten,
unabhängig davon,
welches Gerät darunter eingebunden ist.

Merksatz:

Linux hängt Laufwerke in den Verzeichnisbaum ein.

Groß- und Kleinschreibung

Linux unterscheidet bei Dateinamen zwischen Groß- und Kleinschreibung.

Diese Dateien wären unterschiedlich:

Datei.txt

datei.txt

DATEI.txt

Wichtig:

Befehle,
Pfade
und Dateinamen müssen exakt geschrieben werden.

Merksatz:

Linux ist case-sensitive.

Leerzeichen in Dateinamen

Leerzeichen in Dateinamen können in der Shell Probleme verursachen, wenn sie nicht korrekt behandelt werden.

Beispiel:

Meine Datei.txt

Mögliche Schreibweisen:

"Meine Datei.txt"

oder:

Meine\ Datei.txt

Besser für Server und Skripte:

meine_datei.txt

meine-datei.txt

Merksatz:

Leerzeichen in Dateinamen besser vermeiden,
besonders in Skripten.

Dateiendungen unter Linux

Linux ist weniger abhängig von Dateiendungen als Windows.

Eine Datei kann ausführbar sein, wenn sie die passenden Rechte besitzt, nicht nur, weil sie eine bestimmte Endung hat.

Beispiele:

```
script.sh  
  
backup  
  
programm
```

Trotzdem sind Endungen nützlich, um Dateien leichter zu erkennen.

Beispiele:

```
.txt  
  
.conf  
  
.log  
  
.sh  
  
.tar.gz
```

Merksatz:

Unter Linux sind Rechte wichtiger als Dateiendung.

Wichtige Befehle zu Pfaden und Verzeichnissen

Befehl	Bedeutung
pwd	aktuelles Verzeichnis anzeigen
ls	Inhalt anzeigen
ls -l	ausführliche Liste anzeigen

Befehl	Bedeutung
ls -a	versteckte Dateien anzeigen
cd	Verzeichnis wechseln
cd ..	eine Ebene höher
cd ~	ins Home-Verzeichnis
mkdir	Verzeichnis erstellen
rmdir	leeres Verzeichnis löschen
tree	Verzeichnisbaum anzeigen, falls installiert

Merksatz:

```
pwd,  
ls  
und cd sind Grundbefehle für Navigation.
```

Typische Beispiele

Aktuelles Verzeichnis anzeigen:

```
pwd
```

In das Home-Verzeichnis wechseln:

```
cd ~
```

Nach /etc wechseln:

```
cd /etc
```

Eine Ebene nach oben wechseln:

```
cd ..
```

Inhalt anzeigen:

```
ls
```

Ausführliche Liste anzeigen:

```
ls -l
```

Auch versteckte Dateien anzeigen:

```
ls -a
```

Verzeichnis erstellen:

```
mkdir testordner
```

Merksatz:

Erst wissen,
wo man ist.
Dann Befehl ausführen.

Typische Prüfungsfragen

Frage	Kurzantwort
Was ist /?	Root-Verzeichnis des Dateisystems
Was ist /root?	Home-Verzeichnis des Root-Benutzers
Was ist root?	administrativer Benutzer
Wo liegen viele Konfigurationsdateien?	/etc
Wo liegen viele Logs?	/var/log
Wo liegen Benutzerverzeichnisse?	/home
Was bedeutet ~?	eigenes Home-Verzeichnis
Was bedeutet ..?	übergeordnetes Verzeichnis
Was ist ein absoluter Pfad?	Pfad ab /
Was ist ein relativer Pfad?	Pfad relativ zum aktuellen Verzeichnis
Was ist ein Mountpoint?	Einhängepunkt für ein Dateisystem
Was bedeutet versteckte Datei?	Name beginnt mit Punkt

Typische Prüfungsfallen

Falle	Richtig denken
/ und /root verwechseln	/ ist Dateisystemwurzel, /root ist Root-Home
root und / verwechseln	root ist Benutzer, / ist Verzeichnis
absolute und relative Pfade verwechseln	absolut beginnt mit /
~ als normales Zeichen verstehen	~ steht für Home-Verzeichnis
/tmp dauerhaft nutzen	/tmp ist temporär
/etc löschen oder blind ändern	enthält wichtige Konfiguration
Logs nicht unter /var/log suchen	/var/log ist zentral
Groß- und Kleinschreibung ignorieren	Linux unterscheidet
Leerzeichen in Skripten vergessen	Anführungszeichen oder Escape nötig
Mountpoint nicht verstehen	Dateisystem wird in Verzeichnis eingehängt

IHK-sichere Kurzformulierung

Linux verwendet einen hierarchischen Verzeichnisbaum, der beim Root-Verzeichnis / beginnt. Alle Dateien, Verzeichnisse, Geräte und eingebundenen Dateisysteme befinden sich unterhalb dieses Baums. Ein absoluter Pfad beginnt immer mit /, ein relativer Pfad bezieht sich auf das aktuelle Verzeichnis. Benutzerdateien liegen meist unter /home, systemweite Konfigurationen unter /etc und Logdateien häufig unter /var/log. Das Home-Verzeichnis des administrativen Benutzers root ist /root und darf nicht mit dem Root-Verzeichnis / verwechselt werden. Datenträger werden unter Linux nicht über Laufwerksbuchstaben angesprochen, sondern an Mountpoints in den Verzeichnisbaum eingehängt.

Merksätze

Unter Linux beginnt alles bei /.

/ ist das Root-Verzeichnis.

root ist der administrative Benutzer.

/root ist das Home-Verzeichnis von root.

/home enthält Benutzerverzeichnisse.

/etc enthält Konfigurationen.

/var enthält veränderliche Daten.

/var/log enthält Logs.

/tmp ist temporär.

/usr enthält Programme und Ressourcen.

/dev enthält Gerätedateien.

/proc enthält Prozess- und Kernelinformationen.

/sys enthält System- und Geräteinformationen.

/mnt ist oft für manuelle Mounts.

/media ist oft für Wechselmedien.

/opt ist für optionale Zusatzsoftware.

Absoluter Pfad beginnt mit /.

Relativer Pfad hängt vom aktuellen Verzeichnis ab.

pwd zeigt das aktuelle Verzeichnis.

cd wechselt das Verzeichnis.

. bedeutet aktuelles Verzeichnis.

.. bedeutet übergeordnetes Verzeichnis.

~ bedeutet eigenes Home-Verzeichnis.

Punkt am Anfang bedeutet versteckte Datei.

Linux unterscheidet Groß- und Kleinschreibung.

Laufwerke werden unter Linux eingehängt,
nicht als C: oder D: angezeigt.

Mountpoint = Einhängpunkt im Verzeichnisbaum.