

Systembefehle

ITS-Net-Lin

Sebastian Meisel

12. Dezember 2024

1 Verzeichnisstruktur und Laufwerke

Unixoide Systeme, wie Linux, verwenden eine baumartige Verzeichnisstruktur. Diese beginnt mit dem Verzeichnis /. Dieses Verzeichnis wird auch Root-Verzeichnis genannt. Von dort aus verzweigen sich alle weiteren Verzeichnisse.

```
/
├── bin -> usr/bin      # Programme des Paketmanagers für alle Benutzer.
├── boot                # Bootloader und Dateien für den Bootprozess.
├── dev                 # Schnittstellen zu Geräten über Pseudo-Dateien.
├── etc                 # Systemweite Konfigurationsdateien.
├── home                # Verzeichnisse der Benutzer.
├── lib -> usr/lib       # 32-Bit-Bibliotheken.
├── lib64 -> usr/lib64  # 64-Bit-Bibliotheken.
├── mnt                 # Mountpoint für externe Dateisysteme.
├── opt                 # Große Programme von Drittanbietern.
├── proc                # Pseudo-Dateien mit Systeminformationen.
├── root                # Heimatverzeichnis des Root-Benutzers.
├── run                 # Laufzeitdateien.
├── sbin -> usr/sbin     # Superuser-Programme des Paketmanagers.
├── sys                 # Geräteinformationen als Pseudo-Dateien.
├── tmp                 # Temporäre Dateien, die nach Neustart gelöscht werden.
├── usr                 # Nicht direkt benötigte Systemkomponenten.
└── var                 # Logs, Cache-Dateien und dynamische Inhalte.
```

Anders als bei Windows zeigt ein Verzeichnis nicht an, auf welchem Laufwerk es liegt. In Linux können Laufwerke an beliebigen Stellen in dieser Baumstruktur eingehängt werden.

1.1 /dev

Das Verzeichnis /dev enthält Schnittstellen zu Geräten und stellt diese als Pseudo-Dateien (API) dar. Hier finden sich beispielsweise:

- /dev/sda oder /dev/nvme0n1: Geräte für Festplatten.
- /dev/null: Eine Art "Datenmülleimer", der alle geschriebenen Daten verwirft.
- /dev/tty: Virtuelle Terminals.

1.2 /proc

Das Verzeichnis `/proc` ist ebenfalls ein virtuelles Dateisystem (API), das Laufzeitinformationen über den Kernel und die laufenden Prozesse bereitstellt. Beispiele:

- `/proc/cpuinfo`: Informationen über den Prozessor.
- `/proc/meminfo`: Informationen über den Speicherverbrauch.
- `/proc/[PID]`: Unterverzeichnisse für jeden laufenden Prozess, basierend auf der Prozess-ID (PID).

1.3 /sys

Das Verzeichnis `/sys` ist ein weiteres virtuelles Dateisystem (API), das Laufzeitinformationen über Geräte und Hardware bereitstellt. Es wird vom Kernel bereitgestellt und ermöglicht eine Interaktion mit Hardwarekomponenten. Beispiele:

- `/sys/class`: Klassifizierung von Geräten, z. B. Netzwerkadapter oder Speichergeräte.
- `/sys/devices`: Hierarchische Darstellung der Hardware-Geräte und ihrer Abhängigkeiten.
- `/sys/block`: Informationen über Blockgeräte wie Festplatten und Partitionen.

Das `/sys`-Verzeichnis ergänzt `/proc`, indem es eine strukturierte und oft besser lesbare Darstellung für Hardware-nahe Informationen bietet. Es wird häufig von Systemadministratoren und Entwicklern genutzt, um die Hardware eines Systems zu konfigurieren oder zu überwachen.

1.4 /var

Das Verzeichnis `/var` enthält Dateien, deren Inhalt sich dynamisch ändert, z. B.:

- `/var/log`: System- und Anwendungslogs.
- `/var/spool`: Warteschlangen für Drucker oder Mails.
- `/var/cache`: Zwischenspeicher von Anwendungen.

1.5 /usr

Das Verzeichnis `/usr` ist für Benutzerprogramme und Bibliotheken vorgesehen, die nicht direkt für den Systemstart benötigt werden. Beispiele:

- `/usr/bin`: Programme für alle Benutzer.
- `/usr/lib`: Bibliotheken für diese Programme.
- `/usr/share`: Architekturunabhängige Dateien wie Dokumentationen.

1.6 /usr/local

Das Verzeichnis `/usr/local` ist für Software gedacht, die lokal installiert wurde, ohne den Paketmanager zu verwenden. Dies verhindert Konflikte mit Systemdateien:

- `/usr/local/bin`: Eigene Programme für alle Benutzer.
- `/usr/local/lib`: Eigene Bibliotheken.

1.7 lsblk

Der Befehl `lsblk` listet die verfügbaren Laufwerke (Blockgeräte) und zeigt an, wo sie im Verzeichnisbaum eingehängt sind.

Zum Beispiel könnte das Verzeichnis `/home` auf der zweiten Partition der ersten Festplatte (`sda6`) liegen.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
sda	254:0	0	50G	0	disk	
—sda1	254:1	0	512M	0	part	/boot/efi
—sda2	254:2	0	9.5G	0	part	/
—sda3	254:3	0	3.5G	0	part	/var
—sda4	254:4	0	977M	0	part	[SWAP]
—sda5	254:5	0	680M	0	part	/tmp
—sda6	254:6	0	34.9G	0	part	/home
sr0	11:0	1	1024M	0	rom	

1.8 df

Mit dem Befehl `df` kann überprüft werden, wie viel Speicherplatz auf den Dateisystemen verfügbar oder belegt ist. Mit der Option `-h` (human-readable) werden die Ergebnisse in leicht lesbaren Größen (z. B. MiB oder GiB) angezeigt.

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	947M	0	947M	0%	/dev
tmpfs	195M	1.4M	194M	1%	/run
/dev/vda2	9.3G	4.4G	4.5G	50%	/
tmpfs	974M	0	974M	0%	/dev/shm
tmpfs	5.0M	8.0K	5.0M	1%	/run/lock
/dev/vda3	3.4G	420M	2.8G	13%	/var
/dev/vda6	35G	1.1G	32G	4%	/home
/dev/vda5	652M	124K	605M	1%	/tmp
/dev/vda1	511M	5.9M	506M	2%	/boot/efi
tmpfs	195M	80K	195M	1%	/run/user/1000

Dabei werden auch virtuelle Dateisysteme angezeigt:

- `tmpfs`: Temporäre Dateisysteme im RAM.
- `udev`: Virtuelles Dateisystem für Gerätekommunikation.