

# Dateisystem



# Inhaltsverzeichnis

- [Linux-Dateisystem](#)
- [Die wichtigsten Haupt-Verzeichnisse unter /](#)
- [Weitere Haupt-Verzeichnisse unter /](#)
- [Dateibaum bestehend aus Teilbäumen](#)
- [Dateisystem-Typen](#)
- [Anzeigen der eingehängten Dateisysteme mit df \(disk free\)](#)
- [Einhängen der Dateisysteme beim Booten](#)
- [Key Takeaways](#)

# Linux-Dateisystem

Das Linux-Dateisystem ist eine hierarchische Struktur, die sich von einem einzigen Wurzelverzeichnis aus erstreckt. Das Wurzelverzeichnis wird durch einen Schrägstrich ( / ) dargestellt. Alle anderen Dateien und Verzeichnisse sind unterhalb des Wurzelverzeichnisses angeordnet.

Die Verzeichnisse der ersten Ebene unterhalb des Wurzelverzeichnisses genügen gewissen Unix-Konventionen und -Standards. Hier ein Überblick:

```
hermann@debian:~$ tree -L 1 /
/
├── bin -> usr/bin
├── boot
├── dev
├── etc
├── home
├── initrd.img -> boot/initrd.img-6.1.0-27-amd64
├── initrd.img.old -> boot/initrd.img-6.1.0-26-amd64
├── lib -> usr/lib
├── lib64 -> usr/lib64
├── lost+found
├── media
├── mnt
└── opt
```

```
|— proc
|— root
|— run
|— sbin -> usr/sbin
|— srv
|— sys
|— tmp
|— usr
|— var
|— vmlinuz -> boot/vmlinuz-6.1.0-27-amd64
|— vmlinuz.old -> boot/vmlinuz-6.1.0-26-amd64
```

21 directories, 4 files

# Die wichtigsten Haupt-Verzeichnisse unter `/`

- `/bin` - elementare Programme, die zum Booten und Reparieren des Systems benötigt werden. Diese Programme sind jedoch auch für normale Benutzer verfügbar. Liegen `/` und `/usr` auf demselben Dateisystem, dann ist `/bin` meist ein symbolischer Link auf `/usr/bin`.
- `/dev` - Gerätedateien, die dem Zugriff auf die Hardware dienen
- `/etc` - Konfigurationsdateien für das System und die Anwendungen

- **/home** - Basisverzeichnis der Benutzerverzeichnisse. Jeder Benutzer hat hier ein eigenes Unterverzeichnis /home/\$USER - sein Heimatverzeichnis.
- **/media** - Verzeichnis, in das Wechseldatenträger (USB-Sticks, CD-ROMs etc.) eingehängt werden
- **/opt** - zusätzliche, optionale Software, die nicht zum Betriebssystem gehört
- **/proc** - virtuelles Dateisystem, das Informationen über laufende Prozesse bereitstellt

- `/sbin` - elementare Systemprogramme, die nur von `root` ausgeführt werden können. Liegen `/` und `/usr` auf demselben Dateisystem, dann ist `/sbin` meist ein symbolischer Link auf `/usr/sbin`.
- `/tmp` - temporäre Dateien aller Benutzer



- **/usr** - (Unix System Resources) Programme, Bibliotheken und Dokumentationen für Benutzer. Dieses Verzeichnis enthält in der Regel nur unveränderliche Dateien, die von den Benutzern nur gelesen werden. (Diese Dateien werden nur bei der Installation von Software-Paketen und beim Update verändert.) Ist **/usr** auf einem eigenen Dateisystem, dann kann dieses mit der Option **ro** (read-only) gemountet (eingehängt) werden.
- **/var** - variable Daten, z.B. Logdateien, Druckwarteschlangen, Mailboxen etc. Ist **/var** auf einem eigenen Dateisystem, dann muss dieses Dateisystem lesbar und beschreibbar sein (Option **rw**).

# Weitere Haupt-Verzeichnisse unter

- **/boot** - Dateien, die für den Bootvorgang benötigt werden: Kernel, Initrd (Initial RAM Disk), Bootloader und -konfiguration, EFI etc.
- **/lib** - Bibliotheken, die von den Programmen benötigt werden
- **/lib64** - 64-Bit-Bibliotheken
- **/lost+found** - Verzeichnis für Dateien, die bei einem Systemabsturz verloren gegangen und bei der nächsten Reparatur des Dateisystems gefunden wurden
- **/mnt** - temporäres Verzeichnis für das Einhängen von Dateisystemen

- **/root** - das Heimatverzeichnis des Benutzers **root**
- **/run** - Sperr-Dateien, temporäre Dateien, die beim Booten erstellt werden
- **/srv** - Daten für Dienste, die auf dem System laufen
- **/sys** - virtuelles Dateisystem, das Informationen über die Hardware bereitstellt

# Dateibaum bestehend aus Teilbäumen

Virtuell sieht der Benutzer den Dateibaum als eine einzige große Struktur. In Wirklichkeit besteht der Dateibaum aus mehreren Teilbäumen, die auf verschiedenen Dateisystemen liegen können. Diese Teilbäume werden in leere Verzeichnisse eingehängt. Dieser Vorgang wird als **Mounten** bezeichnet.

Die eingehängten Dateisysteme können entweder physische Dateisysteme (Festplattenpartitionen, USB-Sticks, CD-ROMs, Netzwerk-Dateisysteme etc.) oder virtuelle Dateisysteme (`/dev`, `/proc`, `/sys` etc.) sein.

# Dateisystem-Typen

Die Dateisysteme in Linux können in verschiedene Kategorien eingeteilt werden:

- **Virtuelle Dateisysteme:** Diese Dateisysteme werden vom Linux-Kernel automatisch bereitgestellt. Sie dienen nicht der persistenten Speicherung von Daten.

- **Physische Dateisysteme:** Diese Dateisysteme liegen auf einem physischen Speichermedium und speichern Daten persistent.
  - **Lokale Dateisysteme:** Lokale Massenspeicher (Festplatten, SSDs, USB-Sticks, CD-ROMs)
  - **Netzwerk-Dateisysteme:** Dateisysteme, die von einem File-Server bereitgestellt werden (NFS, CIFS etc.)

# Virtuelle Dateisysteme

Virtuelle Dateisysteme sind Dateisysteme, die nicht auf einem physischen Speichermedium liegen, sondern vom Betriebssystem automatisch bereitgestellt werden. In Linux sind das die Dateisysteme `/proc`, `/sys`, `/dev` etc.

Die virtuellen Dateisysteme sind nicht persistent, d.h. sie werden bei jedem Neustart des Systems neu erstellt. Sie werden deshalb auch nicht "administriert". Sie dienen dem Administrator aber als Informationsquelle über den Zustand des Systems.

Virtuelle Dateisysteme sind nicht im Fokus dieses Kapitels.

# Physische Dateisysteme

Physische Dateisysteme sind Dateisysteme, die auf einem lokalen oder im Netzwerk auf einem entfernten Speichermedium liegen. Die gespeicherten Daten sind persistent, d.h. sie bleiben auch nach einem Neustart des Systems erhalten.

- Bei **lokalen Dateisystemen** sind die Speichermedien direkt mit dem Rechner verbunden (Festplatten, SSDs, USB-Sticks, CD-ROMs). Die Speichermedien werden ggf. in Partitionen aufgeteilt und formatiert, um dann als Teilbaum in den Dateibaum eines Linux-Systems eingehängt zu werden.



- **Netzwerk-Dateisysteme** werden von einem File-Server bereitgestellt und über das Netzwerk eingebunden (NFS, CIFS etc.).

Für den Benutzer ist der Netzwerk-Zugriff auf Dateien und Verzeichnisse transparent. D.h. der Benutzer sieht keinen Unterschied zwischen einem lokalen und einem Netzwerk-Dateisystem. De facto wird jedoch jeder lesende oder schreibende Zugriff auf Dateien und Verzeichnisse über das Netzwerk-Dateisystem an den File-Server weitergeleitet.

# Anzeige der eingehängten Dateisysteme mit **df** (disk free)

Das **df**-Kommando zeigt standardmäßig folgende Informationen zu jedem eingehängten Dateisystem an:

- **Dateisystem** - Gerät oder Dateisystemname - die Quelle, die eingehängt wurde
- **1K-Blöcke** - die Größe des Dateisystems in 1-Kilobyte-Blöcken
- **Benutzt** - die Anzahl der 1K-Blöcke, die vom Dateisystem benutzt werden

- **Verfügbar** - die Anzahl der 1K-Blöcke, die vom Dateisystem noch frei sind
- **Verw%** - der Prozentsatz der benutzten Blöcke
- **Eingehängt auf** - das Verzeichnis, in das das Dateisystem eingehängt wurde

Mit der Option **-T**:

- **Typ** - der Dateisystemtyp (z.B. **ext4**, **vfat**, **nfs**, **cifs**) wird mit der Option **-T** in der 2. Spalte angezeigt.

Physische Dateisysteme liegen auf einem Datenträger (Festplatte, SSD, USB-Stick oder CD-ROM) oder sie sind Netzwerk-Dateisysteme, die von einem File-Server bereitgestellt werden.

Linux verwendet viele virtuelle Dateisysteme, die nicht auf einer physischen Medium gespeichert sind (z.B. `proc`, `sysfs`, `tmpfs`, `devtmpfs`, etc.). Diese Dateisysteme werden im Dateisystembaum eingehängt, um Informationen über das System bereitzustellen.

Die meisten dieser virtuellen Dateisysteme werden von `df` nicht angezeigt. Mit der Option `-a` zeigt `df` alle Dateisysteme (physische und virtuelle) an.

## Optionen von `df`

- `-k` - (default) Anzeige der Größenangaben in Kilobytes
- `-m` - Anzeige der Größenangaben in Megabytes
- `-h` - Anzeige der Größenangaben in human-readable Form (z.B. 1K, 1M, 1G)
- `-T` - Anzeige des Dateisystemtyps in der 2. Spalte
- `-a` - Anzeige aller Dateisysteme, auch aller virtuellen Dateisysteme
- `-t <typ>` - Anzeige nur der Dateisysteme des Typs `<typ>`
- `-x <typ>` - Anzeige aller Dateisysteme außer vom Typ `<typ>`

## df ohne Optionen

```
hermann@debian:~$ df
Dateisystem      1K-Blöcke Benutzt Verfügbar Verw% Eingehängt auf
udev             1975984         0   1975984    0% /dev
tmpfs             400884        552   400332    1% /run
/dev/sda2        18964304  5078268  12897364   29% /
tmpfs            2004412         0   2004412    0% /dev/shm
tmpfs              5120         0      5120    0% /run/lock
/dev/sda1         523244        5984   517260    2% /boot/efi
tmpfs            400880         48   400832    1% /run/user/1000
```

Mit der Option **-a** werden auch viele virtuelle Dateisysteme angezeigt, die von **df** standardmäßig nicht angezeigt werden.

## **df -T** gibt auch den Dateisystemtyp an

```
hermann@debian:~$ df -T
```

Dateisystem	Typ	1K-Blöcke	Benutzt	Verfügbar	Verw%	Eingehängt auf
udev	devtmpfs	1975984	0	1975984	0%	/dev
tmpfs	tmpfs	400884	552	400332	1%	/run
/dev/sda2	ext4	18964304	5078268	12897364	29%	/
tmpfs	tmpfs	2004412	0	2004412	0%	/dev/shm
tmpfs	tmpfs	5120	0	5120	0%	/run/lock
/dev/sda1	vfat	523244	5984	517260	2%	/boot/efi
tmpfs	tmpfs	400880	48	400832	1%	/run/user/1000

**df -x <typ>** schließt Dateisysteme vom Typ **<typ>** aus

```
hermann@debian:~$ df -T -x tmpfs -x devtmpfs
Dateisystem    Typ  1K-Blöcke Benutzt Verfügbar Verw% Eingehängt auf
/dev/sda2      ext4  18964304 5078268  12897364   29% /
/dev/sda1      vfat   523244    5984    517260    2% /boot/efi
```

Nun sind alle virtuellen Dateisysteme ausgefiltert. Es werden nur die physischen Dateisysteme angezeigt.

- **/dev/sda1** ist das EFI-Systempartition, formatiert mit **vfat** und eingehängt in **/boot/efi**.
- **/dev/sda2** ist die root-Partition, formatiert mit **ext4** und eingehängt in **/**.



## Gerätedateien der 1. Festplatte `/dev/sda`

```
hermann@debian:~$ ls -l /dev/sda /dev/sda?  
brw-rw---- 1 root disk 8, 0 24. Nov 17:07 /dev/sda  
brw-rw---- 1 root disk 8, 1 24. Nov 17:07 /dev/sda1  
brw-rw---- 1 root disk 8, 2 24. Nov 17:07 /dev/sda2  
brw-rw---- 1 root disk 8, 3 24. Nov 17:07 /dev/sda3
```

## `/dev/sda3` wird als Swap-Partition verwendet

```
hermann@debian:~$ sudo swapon -s
```

Filename	Type	Size	Used	Priority
/dev/sda3	partition	999420	0	-2

- `/dev/sda` ist das Gerät für die gesamte 1. Festplatte, die wir im Hyper-V-Manager erstellt haben. Bei der Installation von Debian Linux wurde die Festplatte in 3 Partitionen aufgeteilt.
- `/dev/sda1` ist die EFI-Systempartition (Formatierung: `vfat`). Sie enthält den Bootloader und die Boot-Konfiguration.
- `/dev/sda2` ist unsere root-Partition, auf der die Dateien des Betriebssystems und der Benutzer liegen (Formatierung: `ext4`).
- `/dev/sda3` ist die Swap-Partition. Sie wird als virtueller Arbeitsspeicher verwendet und nicht als Dateisystem eingehängt. Mit dem Kommando `swapon -s` wird der Status der Swap-Partition jedoch angezeigt.

**df ... -m** zeigt die Größenangaben in Megabytes

```
hermann@debian:~$ df -T -x tmpfs -x devtmpfs -m
Dateisystem      Typ  1M-Blöcke Benutzt Verfügbar Verw% Eingehängt auf
/dev/sda2        ext4   18520     4960    12596   29% /
/dev/sda1        vfat     511        6      506    2% /boot/
```

**df ... -h** zeigt die Größen "human-readable" mit Angabe der Einheit

```
hermann@debian:~$ df -T -x tmpfs -x devtmpfs -h
Dateisystem      Typ  Größe Benutzt Verf. Verw% Eingehängt auf
/dev/sda2        ext4   19G   4,9G   13G   29% /
/dev/sda1        vfat  511M   5,9M  506M    2% /boot/efi
```

## Alias `dfp` zur Anzeige nur physischer Dateisysteme

```
hermann@debian:~$ alias dfp='df -T -h -x tmpfs -x devtmpfs'
hermann@debian:~$ dfp
```

Dateisystem	Typ	Größe	Benutzt	Verf.	Verw%	Eingehängt auf
/dev/sda2	ext4	19G	5,3G	12G	31%	/
/dev/sda1	vfat	511M	5,9M	506M	2%	/boot/efi

Dieser Alias nimmt die Typ-Spalte in die Ausgabe von `df` auf und filtert die virtuellen Dateisysteme `tmpfs` und `devtmpfs` aus. `dfp` zeigt also nur die physischen Dateisysteme an.

Definieren Sie diesen Alias in `~/.bash_aliases`, sodass er in jeder Shell-Sitzung verfügbar ist.

# Einhängen der Dateisysteme beim Booten

Die **virtuellen Dateisysteme** wie `/proc`, `/sys`, `/dev` etc. werden beim Booten automatisch eingehängt. Dieser Vorgang kann nicht administrativ beeinflusst werden.

Die **physischen Dateisysteme** wie `/`, `/boot`, `/home`, `/var` etc. sind administrativ steuerbar. (Bei der Installation von `debian` sind dies nur die Dateisysteme unter `/` und `/boot/efi`). Das Einhängen dieser Dateisysteme wird in der Datei `/etc/fstab` konfiguriert. Beim Bootvorgang werden die Dateisysteme gemäß dieser Konfiguration automatisch eingehängt.

## `/etc/fstab` - file system table

```
hermann@debian:~$ cat /etc/fstab
# /etc/fstab: static file system information.
# <file system> <mount point> <type> <options> <dump> <pass>

# / was on /dev/sda2 during installation
UUID=1cbc22fc-4dad-4f2f-9e7d-d58329c86015 / ext4 errors=remount-ro 0 1

# /boot/efi was on /dev/sda1 during installation
UUID=BA7C-C0C4 /boot/efi vfat umask=0077 0 1

# swap was on /dev/sda3 during installation
UUID=ff32afed-4797-487a-becc-271b43edff14 none swap sw 0 0

/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
```

Sollen dem System weitere Dateisysteme hinzugefügt werden, so sind diese ebenfalls in der Datei `/etc/fstab` zu konfigurieren. Wenn Sie z.B. eine neue Festplatte oder SSD hinzufügen, dann ist diese zuerst zu partitionieren und jede Partition zu formatieren. Anschließend wird die UUID der Partitionen ermittelt und ein weiterer Eintrag in der Datei `/etc/fstab` hinzugefügt.

Auch Netzwerk-Dateisysteme (vom Typ `nfs` oder `cifs`) können in dieser Datei zum automatischen Einhängen beim Boot konfiguriert werden.

# Key Takeaways

- Das Linux-Dateisystem ist eine hierarchische Struktur, die sich von einem einzigen Wurzelverzeichnis aus erstreckt.
- Die wichtigsten Haupt-Verzeichnisse unter `/`:
  - `/bin` und `/usr/bin` enthalten Programme für alle Benutzer
  - `/sbin` und `/usr/sbin` enthalten Programme, die nur von `root` ausgeführt werden können
  - `/dev` enthält Gerätedateien
  - `/etc` enthält die Konfigurationsdateien des Systems



- ○ `/home` enthält die Benutzerverzeichnisse
- ○ `/tmp` enthält temporäre Dateien aller Benutzer
- ○ `/usr` enthält Programme, Bibliotheken und Dokumentationen für Benutzer
- ○ `/var` enthält variable Daten wie Logdateien, Druckwarteschlangen, Mailboxen etc.
- Das gesamte Dateisystem ist aus mehreren Teilbäumen zusammengesetzt.
- Jeder Teilbaum ist wiederum ein eigenes Dateisystem, das in ein leeres Verzeichnis eingehängt (gemountet) wird.

- Es gibt zwei Arten von Dateisystemen:
  - virtuelle Dateisysteme ( `/dev`, `/proc`, `/sys` etc.): Diese werden vom Betriebssystem bereitgestellt und enthalten Informationen über das System in Form von Dateien, die in Verzeichnissen strukturiert sind.
  - physische Dateisysteme:
    - Massenspeicher (Festplattenpartitionen, USB-Sticks, CD-ROMs etc.)
    - Netzwerk-Dateisysteme (NFS, CIFS etc.)

- Das `df`-Kommando zeigt Informationen zu den gemounteten Dateisystemen an. Es zeigt, aus welchen Teilbäumen der Dateibaum zusammengesetzt ist.
- Das Zusammensetzen des Dateibaums geschieht beim Booten des Systems.
- Das Mounten der virtuellen Dateisysteme ist automatisch und nicht administrativ steuerbar.
- Das Mounten der physischen Dateisysteme ist administrativ steuerbar und wird in der Datei `/etc/fstab` konfiguriert.