

Dateisystem 101

Einführung in Linux-Dateisysteme

von Marius Schwarz

für die (GNU) Linux UserGroup BraunSchweig

Stand November 2016

Dieses Dokument darf als Ganzes frei verteilt werden. Änderungen sind zu markieren und einem Autor zuzuweisen.

Dateisystem 101

Die Linux-Filesystem-Hierarchie

Zunächst einmal, DAS Linuxfilesystem gibt es nicht. Es gibt viele verschiedene Filesysteme, die alle verschiedene Vorteile haben, sei es Robustheit, Geschwindigkeit oder Kernelsupport.

Einige bekannte Filesysteme (FS) sind : ext2fs ext3fs, ext4fs, reiserfs, xfs, btrfs
Windowsuser kennen vermutlich noch das „ntfs“. Dies wird von den meisten Linuxdistributionen auch unterstützt, was einen einfachen Umgang mit Windowsmedien unter Linux erlaubt.

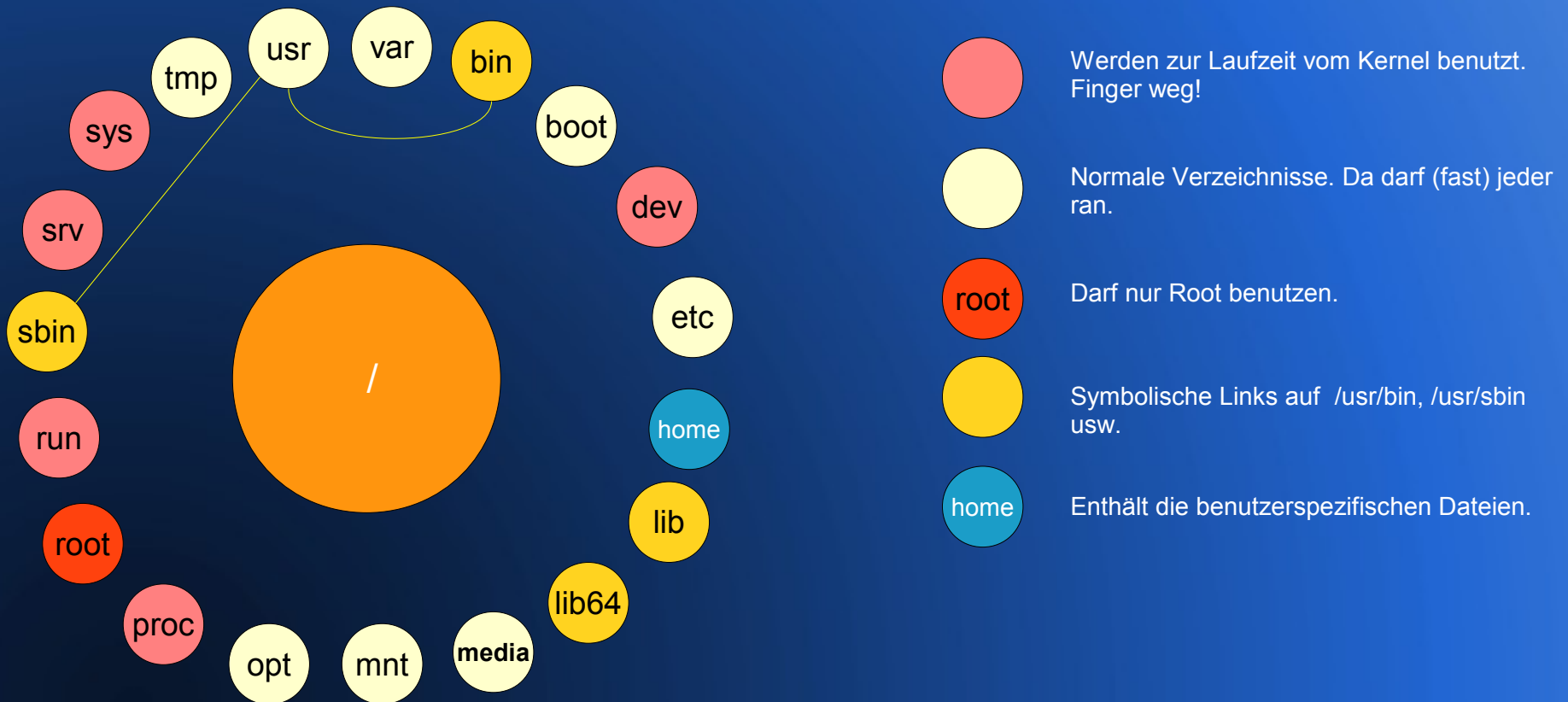
All diesen verschiedenen Filesystemen ist eins gemein: Die Filesystem-Hierarchie

Und in der fängt alles beim Wurzelverzeichnis „/“ (gesprochen Root) an.

„/“ ist die Mutter aller Verzeichnisse auf einem Unix-Filesystem. Egal wieviele Festplatten, DVDROMs, USBSticks oder Netzlaufwerke eingebunden sind, es gibt nur ein gemeinsames Rootverzeichnis.

Die Linux-Filesystem-Hierarchie

Was ist alles im Rootverzeichnis enthalten ?



Die Linux-Filesystem-Hierarchie

Das /boot/ Verzeichnis

Im Boot-Verzeichnis sind die Kernel gelagert. Dort befindet sich auch die Bootkonfiguration:

```
[marius@eve ~]$ ls -ls /boot/
insgesamt 130272
 176 -rw-r--r--. 1 root root 178780  3. Nov 15:50 config-4.8.6-201.fc24.x86_64
 176 -rw-r--r--. 1 root root 178780 11. Nov 17:01 config-4.8.7-200.fc24.x86_64
 176 -rw-r--r--. 1 root root 178780 15. Nov 21:03 config-4.8.8-200.fc24.x86_64
   4 drwx----- 4 root root  4096 29. Okt 12:56 efi
 184 -rw-r--r--. 1 root root 184380  5. Apr 2016 elf-mementest86+-5.01
   4 drwxr-xr-x. 2 root root  4096 29. Okt 12:30 extlinux
   4 drwxr-xr-x. 6 root root  4096 20. Nov 20:06 grub2
37924 -rw----- 1 root root 38832422  1. Jul 2014 initramfs-0-rescue-7e390913b33b4e5ba8f960a9ba97aeee.img
18692 -rw----- 1 root root 19139057  8. Nov 12:03 initramfs-4.8.6-201.fc24.x86_64.img
18660 -rw----- 1 root root 19106565 17. Nov 14:03 initramfs-4.8.7-200.fc24.x86_64.img
18636 -rw----- 1 root root 19082883 20. Nov 20:06 initramfs-4.8.8-200.fc24.x86_64.img
  52 -rw-r--r--. 1 root root  562398 29. Okt 12:43 initrd-plymouth.img
  16 drwx----- 2 root root  16384  1. Jul 2014 lost+found
  180 -rw-r--r--. 1 root root 182704  5. Apr 2016 mementest86+-5.01
3320 -rw----- 1 root root 3396940  3. Nov 15:50 System.map-4.8.6-201.fc24.x86_64
3320 -rw----- 1 root root 3396997 11. Nov 17:01 System.map-4.8.7-200.fc24.x86_64
3320 -rw----- 1 root root 3397313 15. Nov 21:03 System.map-4.8.8-200.fc24.x86_64
 5020 -rwxr-xr-x. 1 root root 5139320  1. Jul 2014 vmlinuz-0-rescue-7e390913b33b4e5ba8f960a9ba97aeee
 6636 -rwxr-xr-x. 1 root root 6794408  3. Nov 15:51 vmlinuz-4.8.6-201.fc24.x86_64
 6636 -rwxr-xr-x. 1 root root 6793608 11. Nov 17:02 vmlinuz-4.8.7-200.fc24.x86_64
 6636 -rwxr-xr-x. 1 root root 6792968 15. Nov 21:04 vmlinuz-4.8.8-200.fc24.x86_64
[marius@eve ~]$
```

Im „initramfs*“ sind alle Dateien enthalten, die ein Kernel zum Booten braucht, der eigentliche Kernel sind die „vmlinuz*“ Dateien.

Die Linux-Filesystem-Hierarchie

Das /boot/ Verzeichnis

Im Verzeichnis „/boot/grub/“ bzw. „/boot/grub2/“ befindet sich die „GRUB“-Bootloader Konfiguration.

Hier wird z.B. entschieden welcher Kernel überhaupt gebootet werden soll und wenn ja, mit welchen Optionen.

Die Grub-Bootloader-Konfiguration wird während des normalen Updates vom Updateprozess erstellt. Normalerweise haben Benutzer daran nichts zu ändern.

Wenn es zu Problemen mit einem Kernel kommt, z.B. das dieser gar nicht hochkommt, dann kann in der grub.cfg der „Default“ Kernel auf einen älteren geändert werden.

Im Initramfs sind z.B. auch die Dateien enthalten, welche die Bootanimation zaubern.

Die Linux-Filesystem-Hierarchie

Die */bin* & */sbin* Verzeichnisse

Früher™ lagen Befehle direkt unter */bin* und Bashscripte unter */sbin*. Später gab es dann „bin“ und „sbin“ nochmal im */usr* Verzeichnis.

Mittlerweile wurde das konsolidiert und */bin* und */sbin* sind nur noch Symbolische Links in die entsprechenden */usr/bin* und */usr/sbin* Verzeichnisse.

Das gleiche Schicksal hat */lib* und */lib64* ereilt. Auch diese Verzeichnisse sind jetzt nur noch unter */usr* zu finden.

Warum wurde daraus symbolische Links gemacht, statt die Verzeichnisse zu entfernen?

Weil man so nur noch einen Ordner mit Befehlen hat, aber alte Scripte noch auf */bin/* referenzieren können, wenn sie einen Befehl aufrufen. So können auch uralte Script und Programm weiterbenutzt werden. Für die Verwalter der Distros vereinfacht dies den Aufwand enorm, da sie nicht an verschiedenen Stellen nachsehen müssen.

Die Linux-Filesystem-Hierarchie

Das /etc Verzeichnis

Im /etc Ordner liegen in verschiedenen Unterverzeichnissen die gesamten Konfigurationsdaten des Systems. Um nur einige zu nennen:

Benutzerdatenbank und Passwortspeicher

X11 Server (Grafik wird unter Linux auch mit Clienten-Server Systemen gemacht)

Netzwerkconfiguration

„SystemD Benutzerkonfigurationen“ und „System V init.d-Dateien“

Hardwareanpassungen mit udevD

Konfigurationsdateien für Paketmanager wie YUM, DNF, Aplitude

Firewalleinstellungen

Apache Webserver

SSH-Server

FTP-Server

usw. usw.

Die Linux-Filesystem-Hierarchie

Das /home-Verzeichnis

Im Home-Verzeichnis hat jeder Benutzer außer Root ein eigenes privates Verzeichnis, auf das nur er allein und der Systemadminaccount Zugriff haben.

Im Home-Verzeichnis eines Benutzers liegen normalerweise nicht nur die Datenfiles, sondern auch die persönlichen Konfigurationsdateien für z.B.

Firefox

Thunderbird

Desktopeinstellungen von Gnome, Cinnamon & Unity

und allen anderen Programmen, die Benutzer einsetzen. Die Verzeichnisstruktur, die beim Anlegen eines Benutzers vorgegeben wird, z.B. für Bilder- und Musik-Ordner, wird von allen Programmen unterstützt und sollte nicht geändert werden.

Ansonsten können Benutzer in Ihrem Home-Verzeichnis machen, was sie wollen.

Konfigurationsdateien liegen üblicherweise in einem versteckten Unterordner.

Die Linux-Filesystem-Hierarchie

Die Utility Verzeichnisse `/media` `/mnt` `/run`

Im Normalfall sind die Verzeichnisse `/media` und `/mnt` leer. Von seiten des Systems her, befinden sich dort nur Informationen, wenn Medien z.b. externe Festplatten eingebunden wurden, oder eine CD/DVD eingelegt wurde.

`/media` wird dann für CD/DVD benutzt,
`/mnt` für Festplatten und ähnliches.

Da immer mehr transportable Datenträger wie USB-Sticks an Linuxsysteme angeschlossen werden, wurde von Seiten der Desktopentwickler das GVFS erfunden.

Das GVFS (Gnome Virtual FileSystem) bindet USB-Sticks und externe Festplatten nicht mehr im `/media` `/mnt` ein, da diese „Mountpoints“ Root vorbehalten sind, der dann auch sicherstellen muß welcher User Rechte bekommt, auf Dateien zuzugreifen.

Das GVFS umgeht das Problem, in dem Medien nach `/run/users/[userid]/gvfs` eingebunden werden. Mit einem „fiesen“ Trick gehören diese Verzeichnisse und Daten dann nur dem aktuellen Desktopbenutzer, so daß andere Benutzer(inkl. root) nicht darauf zugreifen dürfen.

Die Linux-Filesystem-Hierarchie

Die Verzeichnisse `/lib` und `/lib64`

Diese beiden Verzeichnisse enthalten die Systembibliotheken jeweils für 32Bit und 64Bit, was den Zusatz „lib64“ erklärt.

Systembibliotheken sind Dateien, die Programmfunktionen beinhalten, die aus anderen Programmen über eine definierte Schnittstelle heraus aufgerufen werden können.

So müssen Programmteile nicht millionenfach neu programmiert werden und die Bibliotheken können geändert werden, wenn sich z.B. eine Änderung am System ergibt, die berücksichtigt werden muß. Bspw. könnte eine Funktion unter Ubuntu und Fedora den gleichen Namen und gleiche Parameter haben, aber etwas anderes, nämlich etwas distroabhängiges tun, z.B. einen Benutzer anlegen.

Bibliotheken kommen in verschiedenen Versionen und Formen auf allen Betriebssystemen dieser Welt seit den 1960er Jahren zum Einsatz.

Die Linux-Filesystem-Hierarchie

Im /usr-Verzeichnis liegt das System

Das /usr Verzeichnis ist das Herz eines Linuxsystems, da sich dort mittlerweile alle wichtigen Befehle, Bibliotheken und Kerneldateien befinden, die das OS ausmachen.

Es finden sich aber auch die Supportdateien für die verschiedenen Desktops hier, z.B. die MIME Daten, die bestimmen welches Programm gestartet wird, wenn eine Datei vom Type X doppelgeklickt wird.

Kerneldateien z.B. Kernelmodule finden sich nach Kernel aufgeteilt hier:

```
[marius@eve ~]$ ls -la /usr/lib/modules
insgesamt 12
4 drwxr-xr-x. 6 root root 4096  8. Nov 12:05 4.8.6-201.fc24.x86_64
4 drwxr-xr-x. 6 root root 4096 17. Nov 14:05 4.8.7-200.fc24.x86_64
4 drwxr-xr-x. 6 root root 4096 20. Nov 20:08 4.8.8-200.fc24.x86_64
```

Die Linux-Filesystem-Hierarchie

Datenbanken und Logfiles liegen im /var

In Zeiten vor dem Systemd kannte jeder Benutzer die Datei `/var/log/messages`. Dort protokollierte das Linuxsystem sämtliche Meldungen die irgendwie wichtig waren.

Auch heute werden unter `/var/log` noch alle möglichen Logfiles eines Systems gespeichert, z.b. wann welche Updates eingespielt wurden (`/var/log/dnf.rpm.log`) oder wann welcher Benutzer ein- oder ausgeloggt ist (`/var/log/secure`).

In den anderen Verzeichnissen liegen i.d.R. Datenbanken verschiedener Programme z.b. Emaildatenbank in verschiedenste Formaten, die nichts mit Thunderbird zu tun haben, MySQL Datenbank, DEB und RPM Datenbank, Mailserver (wenn man hat).

Es gibt natürlich noch viel mehr was unter `/var` gespeichert wird, denn *var* steht für *various*, also Verschiedenes. Der überwiegende Teil ist aber für Desktopanwender völlig uninteressant, von Logfiles mal abgesehen.

Die Linux-Filesystem-Hierarchie

Die „Kernel“ Verzeichnisse

Die Kernelverzeichnisse `/proc` `/sys` `/srv` `/dev` `/run` sind für normale Desktopbenutzer bestenfalls uninteressant und schlimmstensfalls gefährlich, denn sie erlauben einen direkten Schreib- und Lesezugriff auf den Kernel. Die meisten Optionen sind auch nur zum Auslesen von Daten gedacht z.B. welche Prozesse laufen, welche Partitionen ein Datenträger hat usw. Diese Verzeichnisse sind nur virtuell und werden vom Kernel erzeugt bzw. eingebunden. Stichwort: ProcFS

Als Rootbenutzer kann man aber auch diverse Parameter des Kernels zur Laufzeit eines System ändern und den aktuellen Bedürfnissen anpassen. Z.B. ist es möglich das Verhalten des Auslagerungsspeichers (SWAP) zu beeinflussen um mehr Power zu erhalten oder „mehr“ Speicherplatz im Ram.

Unter `/dev/` finden sich die „Geräte“ wie z.B. `/dev/hda` `/dev/sda` (Festplatten)
`/dev/sr0` (DVD ROM)

Anfängern kann man nur davon abraten dort rumzufummeln, das kann böse ins Auge gehen.

Die Linux-Filesystem-Hierarchie

Temporäre Dateien gehören → /tmp

Temporäre Dateien werden üblicherweise für alle Benutzer im Verzeichnis `/tmp` gespeichert.

Natürlich kann man seine temporären Dateien auch im Homeverzeichnis speichern, aber `/tmp` wird auf Desktopsystemen üblicherweise via dynamischen Ramdrive eingebunden. D.h. statt Festplattenplatz, wird Ram belegt. Das ist viel schneller als die Festplatte und soll ja nur „temporär“ benutzt werden.

Diverse Programme vergessen aber auch mal hinter sich aufzuräumen. Da hat die Ramdisk den Vorteil, daß sie durch den Reboot eines PCs automatisch geleert wird.

Wenn Ihr ein Serverlinux seht, kann es sein, daß die Serverprozesse im `/tmp` eigene Verzeichnisbäume haben, die mit `systemd...` beginnen. Das sind „private“ `/tmp/` Ordner, damit die Dienste sich mit Ihren Tempdateien nicht in die Quere kommen.

Auf die SpezialAttribute des `/tmp` Verzeichnisses gehen wir später mal ein.