

# Die Verzeichnisstruktur von Linux (Debian based)

geschrieben von Andreas Potthoff | 23. April 2023

Dieser Beitrag beschreibt die typische Verzeichnisstruktur für Linux-Systeme, welche auf Debian basieren. Dazu gehören z.B.: Ubuntu, Raspberry Pi OS, Raspbian, Linux Mint, Kali Linux, Kubuntu, Proxmox u.v.m. Eine umfangreiche Liste von Debian based-Linux-Distributionen findet ihr hier: [Distrowatch.com](https://distrowatch.com) – Debian based

Alle elementaren Programme und Konfigurationsdateien liegen in verschiedenen Systemverzeichnissen. Anhand des Filesystem Hierarchie Standard (FHS) wird die Eingliederung der verschiedenen Dateiarten in die Verzeichnisstruktur gezeigt. Der FHS ist eine Richtlinie für die Verzeichnisstruktur unter Unix-ähnlichen Betriebssystemen.

Sie können den folgenden Informationen entnehmen, wo Sie Konfigurationsdateien finden, welche Verzeichnisse Programme enthalten und wo Dokumentationen zu finden sind.

Die folgende tabellarische Übersicht der Verzeichnisstruktur berücksichtigt nicht eine Installation mit gesonderten Partionen (z.B. boot, home, etc.). Außerdem wird Beschreibung der Dateisystem-Hierarchie durch den Konsolenbefehl **man** (manual) gezeigt.

## Tabelle Verzeichnisstruktur

Die *wichtigsten* Verzeichnisse haben folgende Bedeutungen bzw. Inhalte:

Verzeichnis	Beschreibung
/	<b>Wurzelverzeichnis:</b> Das Wurzelverzeichnis und oberste Verzeichnis der Hierarchie.
/bin	<b>Systemprogramme:</b> Die wichtigsten Systemkommandos. Dieses Verzeichnis darf keine Unterverzeichnisse enthalten.
/boot	<b>Bootloader:</b> Die statischen Dateien des Bootloaders und die Kernel.
/dev	<b>Geräte Dateien:</b> Schnittstellen zur Ansteuerung der gesamten Hardware.
/etc	<b>Systemkonfiguration:</b> Die Konfigurationsdateien.
/etc/init.d	<b>Start- und Stopscripte.</b>
/etc/network	<b>Konfigurationsdateien des Netzwerkes.</b> z.B. interfaces
/etc/opt	<b>Konfigurationsdateien für Programme</b> im Verzeichnis /opt
/home/Benutzername	<b>Heimatverzeichnis:</b> Die eigenen Dateien. Für jeden Benutzer finden Sie einen eigenes Verzeichnis mit dem Benutzernamen.
/lib	<b>Bibliotheken:</b> Dynamische Bibliotheken und Kernelmodule die für den Systemstart notwendig sind.
/lib/modules	<b>Kernelmodule</b>
/lost+found	<b>Verloren+Gefunden:</b> Dateien und Dateifragmente die bei der Reparatur eines defekten Dateissystems übrig geblieben sind.
/media	<b>Wechselmedien:</b> Einhängpunkt und Unterverzeichnisse für transportable Medienspeicher (z.B. USB-Stick, externe Festplatte, Floppy, CD-ROM, DVD, u.a).

Verzeichnis	Beschreibung
/mnt	<b>Mountpunkt:</b> Einhängepunkt für ein temporär eingehängtes Dateisystem.
/opt	<b>Optional:</b> Für die manuelle Installation von Programmen die ihre eigenen Bibliotheken mitbringen und nicht zum Standard der Distribution gehören. Unterverzeichnisse sind z.B. minecraft-pi, sonic-pi, vs, Wolfram
/proc	<b>Prozess- und Systeminformationen:</b> Schnittstellen zum aktuell geladenen Kernel und seinen Prozeduren. Dateien lassen sich mittel cat auslesen z.B. "cat proc/version" gibt die aktuelle Kernelversion aus.
/root	<b>Root Heimatverzeichnis:</b> Das Heimatverzeichnis des Superusers (root). Das root-Verzeichnis liegt im Wurzelverzeichnis, falls das home-Verzeichnis auf eine andere Partiton ausgelagert wurde oder ein Zugriff auf home aus irgendeinem Grund nicht möglich ist.
/run	<b>Prozesse:</b> Dateien für laufende Prozesse. Hier befinden sich auch die meisten PID-Dateien (Process Identifier).
/sbin	<b>Systemprogramme (root):</b> Die Systemkommandos zur Systemverwaltung für die Rootrechte benötigt werden. Dieses Verzeichnis darf keine Unterverzeichnisse enthalten.
/srv	<b>Systemdienste:</b> Daten für Systemdienste (in der Regel leer). Dieses Verzeichnis ist noch nicht genau spezifiziert.

Verzeichnis	Beschreibung
/sys	<b>System:</b> Systemweites Geräteverzeichnis welches Informationen und Statistiken über das System und die Komponenten enthält.
/tmp	<b>Temporär:</b> Temporäre Dateien von Programmen und Benutzern. Dieses Verzeichnis wird nach jedem Neustart automatisch geleert.
/usr	<b>UNIX Systemressourcen:</b> Die meisten Systemprogramme, Bibliotheken und installierten Programme.
/usr/bin	Die meisten <b>Benutzerbefehle</b> . Das primäre Verzeichnis für ausführbare Dateien des Systems.
/usr/games	<b>Spiele</b>
/usr/include	<b>Header-Dateien</b> für C-Programme.
/usr/lib	<b>Allgemeine Bibliotheken</b>
/usr/local	<b>Distributionsunabhängige lokale Hierarchie</b> enthält noch einmal die gleiche Verzeichnisstruktur wie das /usr Verzeichnis. Für <b>Programme und Daten</b> gedacht, die von der entsprechenden Distribution des jeweiligen Systems unabhängig installiert worden sind, wie etwa selbstkompilierte oder unabhängig von der Distribution heruntergeladene Programme und Dateien.
/usr/sbin	Weniger wichtige <b>Systemprogramme</b> die im Gegensatz zu /sbin nicht während des Bootvorganges verwendet werden.
/usr/share	<b>Statische architekturunabhänge Dateien</b> z.B. Dokumentationen, Manpages und Wörterbücher.
/usr/src	<b>Source Code</b> zu den Paketen.

Verzeichnis	Beschreibung
/var	<b>Variable Daten:</b> Diese Daten entstehen z.B. im Zuge einer Abarbeitung
/var/backups	<b>Sicherungskopien</b> der Liste der installierten Programme (dpkg).
/var/cache	<b>Zwischenspeicher</b> von Programmen
/var/lib	<b>Variable Statusinformationen</b>
/var/local	<b>Variable Daten</b> im Zusammenhang mit /usr/local
/var/lock	<b>Lock-Dateien</b> zur Prozesssynchronisation
/var/log	<b>Protokolldateien</b>
/var/mail	<b>Mailboxen</b> der Benutzer
/var/opt	<b>Variable Daten</b> der optionalen Programme
/var/run	<b>Dateien zu laufenden Prozessen</b>
/var/spool	Von Programmen <b>gespoolte Daten</b> z.B. Druckaufträge.
/var/tmp	<b>Variable Daten</b> die zwischen Reboots erhalten bleiben.
/var/www	<b>Standardverzeichnis</b> für Inhalte des <b>Webservers</b> .

## Verzeichnisstruktur mit dem Konsolenbefehl tree anzeigen

Um sich einen Überblick der Verzeichnisstruktur zu verschaffen kann man den Konsolenbefehl **tree** benutzen:

Als erstes wechseln wir in das Wurzelverzeichnis und dann lassen wir uns den Verzeichnisbaum mit einer Leveltiefe von 1 anzeigen. Die Leveltiefe kann natürlich auch erweitert werden, indem man die 1 durch die gewünschte Tiefe ersetzt.

```
pi@pi4b:~ $ cd /
```

```
pi@pi4b:/ $ tree -L 1
```

```
.
├── bin
├── boot
├── boot.bak
├── dev
├── etc
├── home
├── lib
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin
├── srv
├── sys
├── tmp
├── usr
└── var
```

20 directories, 0 files

## **Beschreibung der Dateisystem-Hierarchie mit dem Konsolenbefehl man (englisch)**

Wenn Sie folgenden Konsolenbefehl eingeben:

```
man hier
```

erhalten Sie die Beschreibung der Dateisystem-Hierarchie mit folgender Beispielausgabe:

```
HIER(7) Linux Programmer's Manual HIER(7)
```

```
NAME
```

```
hier - description of the filesystem hierarchy
```

## DESCRIPTION

A typical Linux system has, among others, the following directories:

`/` This is the root directory. This is where the whole tree starts.

`/bin` This directory contains executable programs which are needed in single user mode and to bring the system up or repair it.

`/boot` Contains static files for the boot loader. This directory holds only the files which are needed during the boot process. The map installer and configuration files should go to `/sbin` and `/etc`.

`/dev` Special or device files, which refer to physical devices. See `mknod(1)`.

`/etc` Contains configuration files which are local to the machine. Some larger software packages, like X11, can have their own subdirectories below `/etc`. Site-wide configuration files may be placed here or in `/usr/etc`. Nevertheless, programs should always look for these files in `/etc` and you may have links for these files to `/usr/etc`.

`/etc/opt`

Host-specific configuration files for add-on applications installed in `/opt`.

`/etc/sgml`

This directory contains the configuration files for SGML and XML (optional).

`/etc/skel`

When a new user account is created, files from this directory are usually copied into the user's home directory.

`/etc/X11`

Configuration files for the X11 window system (optional).

**/home** On machines with home directories for users, these are usually beneath this directory, directly or not. The structure of this directory depends on local administration decisions.

**/lib** This directory should hold those shared libraries that are necessary to boot the system and to run the commands in the root filesystem.

**/media** This directory contains mount points for removable media such as CD and DVD disks or USB sticks.

**/mnt** This directory is a mount point for a temporarily mounted filesystem. In some distributions, **/mnt** contains subdirectories intended to be used as mount points for several temporary filesystems.

**/opt** This directory should contain add-on packages that contain static files.

**/proc** This is a mount point for the **proc** filesystem, which provides information about running processes and the kernel. This pseudo-filesystem is described in more detail in **proc(5)**.

**/root** This directory is usually the home directory for the root user (optional).

**/sbin** Like **/bin**, this directory holds commands needed to boot the system, but which are usually not executed by normal users.

**/srv** This directory contains site-specific data that is served by this system.

**/tmp** This directory contains temporary files which may be deleted with no notice, such as by a regular job or at system boot up.



`/usr` This directory is usually mounted from a separate partition. It should hold only sharable, read-only data, so that it can be mounted by various machines running Linux.

`/usr/X11R6`

The X-Window system, version 11 release 6 (optional).

`/usr/X11R6/bin`

Binaries which belong to the X-Window system; often, there is a symbolic link from the more traditional `/usr/bin/X11` to here.

`/usr/X11R6/lib`

Data files associated with the X-Window system.

`/usr/X11R6/lib/X11`

These contain miscellaneous files needed to run X; Often, there is a symbolic link from `/usr/lib/X11` to this directory.

`/usr/X11R6/include/X11`

Contains include files needed for compiling programs using the X11 window system. Often, there is a symbolic link from `/usr/include/X11` to this directory.

`/usr/bin`

This is the primary directory for executable programs. Most programs executed by normal users which are not needed for booting or

for repairing the system and which are not installed locally should be placed in this directory.

`/usr/bin/X11`

is the traditional place to look for X11 executables; on Linux, it usually is a symbolic link to `/usr/X11R6/bin`.

`/usr/dict`

Replaced by `/usr/share/dict`.

`/usr/doc`

Replaced by `/usr/share/doc`.

/usr/etc

Site-wide configuration files to be shared between several machines may be stored in this directory. However, commands should

always reference those files using the /etc directory. Links from files in /etc should point to the appropriate files in /usr/etc.

/usr/games

Binaries for games and educational programs (optional).

/usr/include

Include files for the C compiler.

/usr/include/X11

Include files for the C compiler and the X-Window system. This is usually a symbolic link to /usr/X11R6/include/X11.

/usr/include/asm

Include files which declare some assembler functions. This used to be a symbolic link to /usr/src/linux/include/asm.

/usr/include/linux

This contains information which may change from system release to system release and used to be a symbolic link to /usr/src/linux/include/linux to get at operating-system-specific information.

(Note that one should have include files there that work correctly with the current libc and in user space. However, Linux kernel

source is not designed to be used with user programs and does not know anything about the libc you are using. It is very likely

that things will break if you let /usr/include/asm and /usr/include/linux point at a random kernel tree. Debian systems don't do

this and use headers from a known good kernel version, provided in the libc\*-dev package.)

/usr/include/g++

Include files to use with the GNU C++ compiler.

`/usr/lib`

Object libraries, including dynamic libraries, plus some executables which usually are not invoked directly. More complicated programs may have whole subdirectories there.

`/usr/lib/X11`

The usual place for data files associated with X programs, and configuration files for the X system itself. On Linux, it usually is a symbolic link to `/usr/X11R6/lib/X11`.

`/usr/lib/gcc-lib`

contains executables and include files for the GNU C compiler, `gcc(1)`.

`/usr/lib/groff`

Files for the GNU groff document formatting system.

`/usr/lib/uucp`

Files for `uucp(1)`.

`/usr/local`

This is where programs which are local to the site typically go.

`/usr/local/bin`

Binaries for programs local to the site.

`/usr/local/doc`

Local documentation.

`/usr/local/etc`

Configuration files associated with locally installed programs.

`/usr/local/games`

Binaries for locally installed games.

`/usr/local/lib`

Files associated with locally installed programs.

`/usr/local/include`

Header files for the local C compiler.

`/usr/local/info`

Info pages associated with locally installed programs.

`/usr/local/man`

Man pages associated with locally installed programs.

`/usr/local/sbin`

Locally installed programs for system administration.

`/usr/local/share`

Local application data that can be shared among different architectures of the same OS.

`/usr/local/src`

Source code for locally installed software.

`/usr/man`

Replaced by `/usr/share/man`.

`/usr/sbin`

This directory contains program binaries for system administration which are not essential for the boot process, for mounting `/usr`, or for system repair.

`/usr/share`

This directory contains subdirectories with specific application data, that can be shared among different architectures of the same OS. Often one finds stuff here that used to live in `/usr/doc` or `/usr/lib` or `/usr/man`.

`/usr/share/dict`

Contains the word lists used by spell checkers.

`/usr/share/doc`

Documentation about installed programs.

`/usr/share/games`

Static data files for games in /usr/games.

/usr/share/info

Info pages go here.

/usr/share/locale

Locale information goes here.

/usr/share/man

Manual pages go here in subdirectories according to the man page sections.

/usr/share/man/<locale>/man[1-9]

These directories contain manual pages for the specific locale in source code form. Systems which use a unique language and code

set for all manual pages may omit the <locale> substring.

/usr/share/misc

Miscellaneous data that can be shared among different architectures of the same OS.

/usr/share/nls

The message catalogs for native language support go here.

/usr/share/sgml

Files for SGML and XML.

/usr/share/terminfo

The database for terminfo.

/usr/share/tmac

Troff macros that are not distributed with groff.

/usr/share/zoneinfo

Files for timezone information.

/usr/src

Source files for different parts of the system, included with some packages for reference purposes. Don't work here with your own

projects, as files below /usr should be read-only except when

installing software.

`/usr/src/linux`

This was the traditional place for the kernel source. Some distributions put here the source for the default kernel they ship.

You should probably use another directory when building your own kernel.

`/usr/tmp`

Obsolete. This should be a link to `/var/tmp`. This link is present only for compatibility reasons and shouldn't be used.

`/var` This directory contains files which may change in size, such as spool and log files.

`/var/adm`

This directory is superseded by `/var/log` and should be a symbolic link to `/var/log`.

`/var/backups`

Reserved for historical reasons.

`/var/cache`

Data cached for programs.

`/var/catman/cat[1-9]` or `/var/cache/man/cat[1-9]`

These directories contain preformatted manual pages according to their man page section. (The use of preformatted manual pages is deprecated.)

`/var/cron`

Reserved for historical reasons.

`/var/lib`

Variable state information for programs.

`/var/local`

Variable data for `/usr/local`.

`/var/lock`

Lock files are placed in this directory. The naming convention for device lock files is LCK..<device> where <device> is the device's name in the filesystem. The format used is that of HDU UUCP lock files, that is, lock files contain a PID as a 10-byte

ASCII decimal number, followed by a newline character.

/var/log

Miscellaneous log files.

/var/opt

Variable data for /opt.

/var/mail

Users' mailboxes. Replaces /var/spool/mail.

/var/msg

Reserved for historical reasons.

/var/preserve

Reserved for historical reasons.

/var/run

Run-time variable files, like files holding process identifiers (PIDs) and logged user information (utmp). Files in this directory are usually cleared when the system boots.

/var/spool

Spooled (or queued) files for various programs.

/var/spool/at

Spooled jobs for at(1).

/var/spool/cron

Spooled jobs for cron(8).

/var/spool/lpd

Spooled files for printing.

/var/spool/mail

Replaced by /var/mail.

/var/spool/mqueue  
Queued outgoing mail.

/var/spool/news  
Spool directory for news.

/var/spool/rwho  
Spooled files for rwhod(8).

/var/spool/smail  
Spooled files for the smail(1) mail delivery program.

/var/spool/uucp  
Spooled files for uucp(1).

/var/tmp  
Like /tmp, this directory holds temporary files stored for an unspecified duration.

/var/yp  
Database files for NIS.

---

# **Umfrage: Welche Linux Distributionen hast Du verwendet?**

geschrieben von Andreas Potthoff | 23. April 2023

## **Umfrage:**

**Welche Linux Distributionen hast Du verwendet?**



- ☐ Alpine
- ☐ Arch
- ☐ CentOS
- ☐ Debian
- ☐ Elementary OS
- ☐ Fedora
- ☐ FreeBSD
- ☐ Gentoo
- ☐ Kali
- ☐ Manjaro
- ☐ Mint
- ☐ MX Linux
- ☐ open Mandriva
- ☐ openELEC
- ☐ openSUSE
- ☐ openWrt
- ☐ Puppy
- ☐ Raspberry Pi OS
- ☐ Red Hat
- ☐ Slackware
- ☐ Tiny Core
- ☐ Ubuntu (+ Derivate)
- ☐ \_Andere

Abstimmen

Ergebnisse anzeigen



Wird geladen ...

---

## Abstimmungsergebnisse:

Welche Linux Distributionen hast Du verwendet?

- Debian (25%, 6 Stimmen)
- Kali (25%, 6 Stimmen)

- Ubuntu (+ Derivate) (21%, 5 Stimmen)
- Raspberry Pi OS (17%, 4 Stimmen)
- Gentoo (8%, 2 Stimmen)
- Fedora (4%, 1 Stimmen)
- Elementary OS (0%, 0 Stimmen)
- MX Linux (0%, 0 Stimmen)
- openWrt (0%, 0 Stimmen)
- FreeBSD (0%, 0 Stimmen)
- Tiny Core (0%, 0 Stimmen)
- open Mandriva (0%, 0 Stimmen)
- Red Hat (0%, 0 Stimmen)
- Manjaro (0%, 0 Stimmen)
- Puppy (0%, 0 Stimmen)
- openELEC (0%, 0 Stimmen)
- openSUSE (0%, 0 Stimmen)
- CentOS (0%, 0 Stimmen)
- Slackware (0%, 0 Stimmen)
- Alpine (0%, 0 Stimmen)
- Arch (0%, 0 Stimmen)
- Mint (0%, 0 Stimmen)
- \_Andere (0%, 0 Stimmen)

Wähler insgesamt: 6

Abstimmen



Wird geladen ...