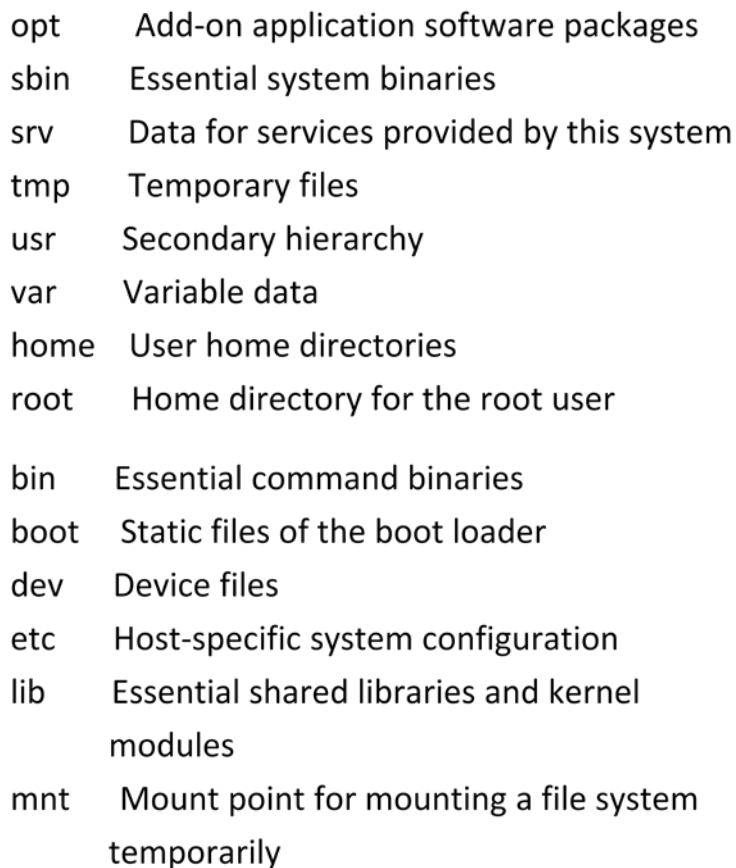


Filesystem Hierarchy Standard



File System Organization

“Like that legacy operating system, the files on a Linux system are arranged in what is called a *hierarchical directory structure*. This means that they are organized in a tree-like pattern of directories (called folders in other systems), which may contain files and other directories. The first directory in the file system is called the *root directory*. The root directory contains files and subdirectories, which contain more files and subdirectories and so on and so on.

[...]

One important difference between the legacy operating system and Unix/Linux is that Linux does not employ the concept of drive letters. While drive letters split the file system into a series of different trees (one for each drive), **Linux always has a single tree**. Different storage devices may contain different branches of the tree, but there is always a single tree.”¹

Command line interface :

“Since a command line interface cannot provide graphic pictures of the file system structure, it must have a different way of representing it. Think of the file system tree as a maze, and you are standing in it. At any given moment, you stand in a single directory. Inside that directory, you can see its files and the pathway to its parent directory and the pathways to the subdirectories of the directory in which you are standing” “The directory you are standing in is called the *working directory*. “ ...”When you first log on to a Linux system, the working directory is set to your home directory” “On most systems, your home directory will be called /home/your_user_name, but it can be anything according to the whims of the system administrator”¹

```
User1@ibt011:~>
```

Command pwd

“To find the name of the working directory, use the **pwd** command.”¹

```
user1@ibt011:~>pwd
```

```
/home/user1
```

Navigation through the filesystem:

cd

“To change your working directory (where you are standing in the maze) you use the **cd** command. To do this, type **cd** followed by the *pathname* of the desired working directory. A *pathname* is the route you take along the branches of the tree to get to the directory you want.

Pathnames can be specified in one of two different ways; *absolute pathnames* or *relative pathnames*¹

Absolute pathnames

“Absolute pathname begins with the root directory and follows the tree branch by branch until the path to the desired directory or file is completed”¹

Example for change directory by absolute pathway:

```
cd /home/user1/bin
```

. dot means the current directory

.. means the upper tree directory

Relative pathnames

The relative pathname is related to the directory you are standing in. Therefore the dot operators are important:

. = one dot means the current directory

.. = two dots means one lower tree directory

For example you standing in the home directory of user one and want to:

- Navigate into the upper directory “home:” `cd ..`
- Stay in directory: `cd .`
- Navigate into the directory /home/ user2: `cd ../usr2`
- Navigate into the directory /home/usr1/bin/test: `cd bin/test`

(The dot operator is not very imported for navigating in the directory but for example for starting a program which is not in the search directory, that means which cannot be started without declaring the directory in which the program is located. If you are standing in the directory where the program is located you can start it for example by:

```
./PROGRAMNAME )
```

“ls - list files

To list the files in the working directory, use the `ls` command.

Format

`ls [options] [file-list]`

Summary

Lists information about one or more files. Lists alphabetically unless you specify something else.

[file-list] could be any directory path or files you specify. `ls` lists the working directory if you don't give it this argument.

Options

-a All

files, includes invisible files (those that start with a period)

-l Long, displays extra information about the files.

-al (-a and -l)

-F Appends a "/" at the end of a directory listing and a " " at the end of an executable listing. Makes the output of ls much more readable. Combine with --color.

--color Color, this command adds colors to the output of ls. A very useful option, make an alias for it!!³

Operators and other helpful thinks		
[command] &	It is possible to reuse the console	firefox &
[command] && [command]	Commands are executed in rotation	firefox && shutdown - h now
[command1] && [command2]	Commands are executed in rotation, stdout of command1 is stdin of command2	ls grep *.txt
[command] *	The command relates to all filenames	cp /home/usr1/* /home/usr2
[command] g*	The command relates to all filenames that begin with the character "g"	cp /home/usr1/g* /home/usr2
[command] b*.txt	The command relates to all filenames that begin with the character "b" and end with the characters ".txt"	cp /home/usr1/b*.txt /home/usr2
[command] >[file]	Deflect the stdout of the command into a file	echo "hallo">hallo.txt
strg + C	Cancel current run	strg + C
man [program]	Display the help of the programm	man firefox
directory, data file commands		
cd	Change the directory	
cd [diretory path]	Change the directory to directory path	cd /home/user
cd ..	Change to directory one level higher	cd ../user2
cd /	Change to root directory	cd /
cd	Change to root directory	cd
cd -	Change to previous directory	cd -

<u>cp</u>	Copy	
cp [file][new file]	Copy file and save the file with new name in the same directory	cp123.txt 1234.txt
cp [file][directory]	Copy file into directory (given by directory path)	cp 123.txt /home/user
cp -r [directory] [directory]	Copy directory into directory	cp123.txt 1234.txt
mv	Move	
mv [file][directory]	Move a file into directory	mv 123.txt /home mv /123
mv [file1][file2]	Change file name from [file1] into [file2]	mv 123.txt 456.txt
rm	Delete	
rm [file]	Delete file	rm /tmp/123.txt ;
rm -rf [directory]	Delete all undergiven directory	rm -rf /tmp/
mkdir	Create a directory	mkdir /home/test
rmdir	Delete a directory	rmdir /home/test
ls	Display content of current/specified directory	ls (/home/test)
ls -l	Listing with more details	ls -l (/home/test)
ls -la	Listing with all details	ls -la (/home/test)
pwd	Show the current directory	pwd
touch [file]	Create empty file	touch 123.txt
which [Prog]	Search the program	whereis Firefox
find (or grep) [directory][file]	Search file in directory	find (or.grep) ../ 12*.txt
grep [KEY] [DATEI]	Search word in file	grep Haus 123.txt
find [directory] -type f -exec grep -l "word" {} \;	Search word in all files in directory find [directory] -type f -> find all files starting from [directory] find -exec -> for each item found, do the	find . -type f -exec grep -l "flux" {} \;

	following command. Where you normally place the filename in you command, put {} . grep -l "word" something -> returns the filename if string "word" is found in file something	
locate [DATEI]	Search for file in the database	locate 123.txt
updatedb	Aktualisiert die Datenbank	updatedb
which	Zeigt an, wo sich ein Programm befindet	which firefox
unzip[Datei]	Entpackt eine Datei	unzip Haus.zip
echo [Text]	Write text into display	echo Hallo
ifconfig	Show infos about the network	ifconfig
ping [Rechner]	Check connection to other PC's	ping 192.168.0.1
ps aux	Display all running processes	ps aux
iwconfig	Show info about WLAN	

Vgl.^{2,3}

Sources: ¹http://linuxcommand.org/learning_the_shell.php

²<http://www.pc-erfahrung.de/linux/linux-befehle.html>

³<http://ss64.com/bash/>

For more information and other commands see ³<http://ss64.com/bash/>