**Important Linux Commands**

This section gives insight into the most important commands of your SuSE Linux system. Along with the individual commands, parameters are listed and, where appropriate, a typical sample application is introduced. To learn more about the various commands, it is usually possible to get additional information with the **man** program followed by the name of the command, for example, **man ls**.

In these [*manual pages*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#manpage), move up and down with **PgUp** and **PgDn** and move between the beginning and the end of a document with **Home** and **End**. End this viewing mode by pressing **Q**. Learn more about the **man** command itself with **man**.

There are many more commands than listed in this chapter. For information about other commands or more detailed information, we recommend the O'Reilly publication Linux in a Nutshell. In the following overview, the individual command elements are written in different typefaces.

* The actual command is always printed as **command**. Without this, nothing can function.
* Options without which the respective program cannot function are printed in italics.
* Further details, like file names, which must be passed to a command for correct functioning, are written in the Courier font.
* Specifications or parameters that are not required are placed in [brackets].

Adjust possible specifications to your needs. It makes no sense to write **ls file(s)**, if no file named file(s) actually exists. You can usually combine several parameters, for example, by writing **ls -la** instead of **ls -l -a**.

### File Commands

#### File Administration

**ls** [option(s)] [file(s)]

If you run **ls** without any additional parameters, the program will list the contents of the current directory in short form.

-l

detailed list

-a

displays hidden files

**cp** [option(s)] sourcefile targetfile

Copies sourcefile to targetfile.

-i

Waits for confirmation, if necessary, before an existing targetfile is overwritten

-r

Copies recursively (includes subdirectories)

**mv** [option(s)] sourcefile targetfile

Copies sourcefile to targetfile then deletes the original sourcefile.

-b

Creates a backup copy of the sourcefile before moving

-i

Waits for confirmation, if necessary, before an existing targetfile is overwritten

**rm** [option(s)] file(s)

Removes the specified files from the file system. Directories are not removed by **rm** unless the option -r is used.

-r

Deletes any existing subdirectories

-i

Waits for confirmation before deleting each file.

**ln** [option(s)] sourcefile targetfile

Creates an internal [*link*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#link) from the sourcefile to the targetfile, under a different name. Normally, such a link points directly to the sourcefile on one and the same file system. However, if **ln** is executed with the -s option, it creates a symbolic link that only points to the directory where the sourcefile is located, thus enabling linking across file systems.

-s

Creates a symbolic link

**cd** [options(s)] [directory]

Changes the current directory. **cd** without any parameters changes to the user's home directory.

**mkdir** [option(s)] directoryname

Creates a new directory.

**rmdir** [option(s)] directoryname

Deletes the specified directory, provided it is already empty.

**chown** [option(s)] username.group file(s)

Transfers the ownership of a file to the user with the specified user name.

-R

Changes files and directories in all subdirectories.

**chgrp** [option(s)] groupname file(s)

Transfers the group ownership of a given file to the group with the specified group name. The file owner can only change group ownership if a member of both the existing and the new group.

**chmod** [options] mode file(s)

Changes the access permissions.

The mode parameter has three parts: group, access, and access type. group accepts the following characters:

u

user

g

group

o

others

For access, access is granted by the + symbol and denied by the - symbol.

The access type is controlled by the following options:

r

read

w

write

x

eXecute — executing files or changing to the directory.

s

Set uid bit — the application or program is started as if it were started by the owner of the file.

**gzip** [parameters] file(s)

This program compresses the contents of files, using complex mathematical algorithms. Files compressed in this way are given the extension .gz and need to be uncompressed before they can be used. To compress several files or even entire directories, use the **tar** command.

-d

decompresses the packed gzip files so they return to their original size and can be processed normally (like the command **gunzip**).

**tar** options archive file(s)

The **tar** puts one file or (usually) several files into an archive. Compression is optional.

**tar** is a quite complex command with a number of options available. The most frequently used options are:

-f

Writes the output to a file and not to the screen as is usually the case

-c

Creates a new tar archive

-r

Adds files to an existing archive

-t

Outputs the contents of an archive

-u

Adds files, but only if they are newer than the files already contained in the archive

-x

Unpacks files from an archive (*extraction*)

-z

Packs the resulting archive with **gzip**

-j

Compresses the resulting archive with **bzip2**

-v

Lists files processed

The archive files created by **tar** end with .tar. If the tar archive was also compressed using **gzip**, the ending is .tgz or .tar.gz. If it was compressed using **bzip2**, .tar.bz2.

Application examples can be found in Section�[“Archives and Data Compression”](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/ch24s02.html#sec:bash.tar).

**locate** pattern(s)

The locate command can find in which directory a specified file is located. If desired, use [*wild cards*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#wildcard) to specify file names. The program is very speedy, as it uses a database specifically created for the purpose (rather than searching through the entire file system). This very fact, however, also results in a major drawback: locate is unable to find any files created after the latest update of its database.

The database can be generated by root with **updatedb**.

**updatedb** [options(s)]

This command performs an update of the database used by **locate**. To include files in all existing directories, run the program as root. It also makes sense to place it in the background by appending an ampersand (&), so you can immediately continue working on the same command line (**updatedb &**).

**find** [option(s)]

The **find** command allows you to search for a file in a given directory. The first argument specifies the directory in which to start the search. The option **-name** must be followed by a search string, which may also include [*wild cards*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#wildcard). Unlike **locate**, which uses a database, **find** scans the actual directory.

#### Commands to Access File Contents

**cat** [option(s)] file(s)

The **cat** command displays the contents of a file, printing the entire contents to the screen without interruption.

-n

Numbers the output on the left margin

**less** [option(s)] file(s)

This command can be used to browse the contents of the specified file. Scroll half a screen page up or down with **PgUp** and **PgDn** or a full screen page down with **Space**. Jump to the beginning or end of a file using **Home** and **End**. Press **Q** to exit the program.

**grep** [option(s)] searchstring filenames

The grep command finds a specific searchstring in the specified file(s). If the search string is found, the command displays the line in which the searchstring was found along with the file name.

-i

Ignores case

-l

Only displays the names of the respective files, but not the text lines

-n

Additionally displays the numbers of the lines in which it found a hit

-l

Only lists the files in which searchstring does not occur

**diff** [option(s)] file1 file2

The **diff** command compares the contents of any two files. The output produced by the program lists all lines that do not match.

This is frequently used by programmers who need only send their program alterations and not the entire source code.

-q

Only reports whether the two given files differ

#### File Systems

**mount** [option(s)] [<device>] mountpoint

This command can be used to mount any data media, such as hard disks, CD-ROM drives, and other drives, to a directory of the Linux file system.

-r

mount read-only

-t filesystem

Specifies the file system. The most common are ext2 for Linux hard disks, msdos for MS-DOS media, vfat for the Windows file system, and iso9660 for CDs.

For hard disks not defined in the file /etc/fstab, the device type must also be specified. In this case, only root can mount. If the file system should also be mounted by other users, enter the option user in the appropriate line in the /etc/fstab file (separated by commas) and save this change. Further information is available in mount.

**umount** [option(s)] mountpoint

This command unmounts a mounted drive from the file system. To prevent data loss, run this command before taking a removable data medium from its drive. Normally, only root is allowed to run the commands **mount** and **umount**. To enable other users to run these commands, edit the /etc/fstab file to specify the option user for the respective drive.

### System Commands

#### System Information

**df** [option(s)] [directory]

The **df** (disk free) command, when used without any options, displays information about the total disk space, the disk space currently in use, and the free space on all the mounted drives. If a directory is specified, the information is limited to the drive on which that directory is located.

-H

shows the number of occupied blocks in gigabytes, megabytes, or kilobytes — in human-readable format

-t

Type of file system (ext2, nfs, etc.)

**du** [option(s)] [path]

This command, when executed without any parameters, shows the total disk space occupied by files and subdirectories in the current directory.

-a

Displays the size of each individual file

-h

Output in human-readable form

-s

Displays only the calculated total size

**free** [option(s)]

The command **free** displays information about RAM and swap space usage, showing the total and the used amount in both categories.

-b

Output in bytes

-k

Output in kilobytes

-m

Output in megabytes

**date** [option(s)]

This simple program displays the current system time. If run as root, it can also be used to change the system time. Details about the program are available in date.

#### Processes

**top** [options(s)]

top provides a quick overview of the currently running [*processes*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#process). Press **H** to access a page that briefly explains the main options to customize the program.

**ps** [option(s)] [process ID]

If run without any options, this command displays a table of all *your own* programs or processes — those you started. The options for this command are not preceded by hyphen.

aux

Displays a detailed list of all processes, independent of the owner.

**kill** [option(s)] process ID

Unfortunately, sometimes a program cannot be terminated in the normal way. However, in most cases, you should still be able to stop such a runaway program by executing the **kill** command, specifying the respective process ID (see **top** and **ps**).

**kill** sends a *TERM* signal that instructs the program to shut itself down. If this does not help, the following parameter can be used:

-9

Sends a *KILL* signal instead of a *TERM* signal, with which the process really is *annihilated* by the operating system. This brings the specific processes to an end in almost all cases.

**killall** [option(s)] processname

This command is similar to **kill**, but uses the process name (instead of the process ID) as an argument, causing all processes with that name to be killed.

#### Network

**ping** [option(s)] host name|IP address

The ping command is the standard tool for testing the basic functionality of TCP/IP networks. It sends a small data packet to the destination host, requesting an immediate reply. If this works, ping displays a message to that effect, which indicates that the network link is basically functioning.

-c

number Determines the total number of packages to send and ends after they have been dispatched. By default, there is no limitation set.

-f

*flood ping*: sends as many data packages as possible. A popular means, reserved to root, to test networks.

-i

value Specifies the interval between two data packages in seconds. Default: one second

**nslookup**

The Domain Name System resolves domain names to IP addresses. With this tool, send queries to information servers (DNS servers).

**telnet** [option(s)] host name or IP address

Telnet is actually an Internet protocol that enables you to work on remote hosts across a network. telnet is also the name of a Linux program that uses this protocol to enable operations on remote computers.

WARNING: Do not use telnet over a network on which third parties can *eavesdrop.* Particularly on the Internet, use encrypted transfer methods, such as **ssh**, to avoid the risk of malicious misuse of a password (see the man page for **ssh**).

#### Miscellaneous

**passwd** [option(s)] [username]

Users may change their own passwords at any time using this command. Furthermore, the administrator root can use the command to change the password of any user on the system.

**su** [option(s)] [username]

The **su** command makes it possible to log in under a different user name from a running session. When using the command without specifying a user name, you will be prompted for the root password. Specify a user name and the corresponding password to use the environment of the respective user. The password is not required from [*root*](https://www-uxsup.csx.cam.ac.uk/pub/doc/suse/suse9.0/userguide-9.0/go01.html#root), as root is authorized to assume the identity of any user.

**halt** [option(s)]

To avoid loss of data, you should always use this program to shut down your system.

**reboot** [option(s)]

Does the same as **halt** with the difference that the system performs an immediate reboot.

**clear**

This command cleans up the visible area of the console. It has no options.