

# Linux Reference Page

Version 3, January 2013

## Logging in

CTRL+ALT+F1 through F6	Virtual (text mode) terminals 1 through 6.
CTRL+ALT+F7	First graphic terminal (desktop manager).
su	Log in as superuser.
su name	Log in as user <b>name</b> .
su - name	Same as above, but assuming the same environment that user would find.

## Exiting

exit	Log out of a session (also <b>CTRL+D</b> ).
shutdown -h -P now	Turn the system off now, and don't reboot.
shutdown -h -P 11:34	Same as above, but at that given time.
halt	Same as <b>shutdown -h -P</b> .
poweroff	Same as above.
reboot	Restart the computer.

## Getting help and info

man name	Show the manual for that program. or important system file. <b>Q</b> quits.
apropos name	List man pages related to that <b>name</b> .
info name	More complete and updated info than <b>man</b> .
TAB	Autocompletion. Double-tapping lists choices.
SHIFT+PG UP	Scroll up in a terminal to read previous text.
SHIFT+PG DOWN	Scroll down.
CTRL+W	Erase the current word in a terminal's input.
date	Print current date and time.
cal	Print a calendar for this month.
uname	Print the name of the machine.
uname -r	Print the running kernel.
uname -a	Print the complete info about the machine.
cat /proc/cpuinfo	Print info about the CPU.
cat /proc/meminfo	Print info about the installed memory.
lspci	List all PCI-connected devices.
lsusb	List all USB-connected devices.
df	Show all filesystems and disk usage.
du name	Show size info about a directory.
du -h name	Same as above, but with human-readable units.
free	Show RAM and swap usage.
uptime	Show the computer's uptime and load average.
dmesg	Print system log (/var/log/syslog).

## Users

whoami	Print user name.
who	Info about active users.
who am i	Same as above but just for you.
w	More detailed info about all active users.
users	Print all users logged in this machine.
adduser name	Add a new user with that <b>name</b> .
deluser name	Remove the user with that <b>name</b> .
passwd name	Change the password of an user.
write name	Send messages to another user ( <b>CTRL+D</b> quits).
mesg y	Enable the reception of messages via <b>write</b> .
mesg n	Disable the reception of messages via <b>write</b> .
mesg	Print the current state (if you are able to receive).

## Environment

printenv	Print all environment variables.
echo \$NAME	Print the environment variable <b>NAME</b> .
export NAME=value	Create an environment variable.
env NAME=value name	Run a program in a custom environment.
alias name='command'	Create a shortcut for a command.

The variable **\$PATH** stores the locations where executables will be sought when invoking a command. To add a new directory:

```
export PATH=$PATH:/new/directory
```

## Directories

pwd	Print current working directory.
cd	Change to your HOME directory.
cd /usr/lib	Change to a directory using its full path.
cd lib	Change to a directory using its relative path.
cd ..	Change to the parent directory.
cd -	Return to the previous directory.
mkdir name	Create a new directory with that <b>name</b> .
cp -r name1 name2	Recursively copy a directory to another.
mv name1 name2	Move an empty directory into another.
rmdir name	Remove an empty directory.
rm -r name	Same as above.
rm -rf name	Force removal of a non-empty directory.

## Files

In Linux, directories are considered files too, and extensions do not matter. File names starting with a dot means the item is hidden.

ls	List contents of current directory.
ls -la	Show more details, including hidden items.
touch name	Create a given file, or update its access info.
nano name	Edit a file as text.
cat name	Print the full contents of a file.
more name	Same as above but interactive, for long files.
less name	Same as above but able to go down and forth.
head -n X name	Print the first X lines of a file (10 is default).
tail -n X name	Print the last X lines of a file (10 is default).
tail -f name	Keep printing as the file grows, good for logs.
cp name1 name2	Copy a file to that destination/name.
mv name1 name2	Change the name of a file, or move it there.
rm name	Remove (not trash) a file.
rm -f name	Same as above, but forcing (ignores warnings).
ln -s name1 name2	<b>name1</b> will be a symbolic link to file <b>name2</b> .
tar cf name files	Create a tar container with one or more files.
tar xf name	Extract the contents of a tar.
tar xjf name	Extract the contents of a bzip2 compressed tar.
tar xzf name	Extract the contents of a Gzip compressed tar.
name < file	That file will be fed as input for the program.
name > file	Save output to that file, overwriting if exists.
name >> file	Same as above, but will append, not overwrite.
name << string	The program will process any input you type next, until that <b>string</b> appears in a line.

## Patterns and RegEx

*Regular expressions* contain one or more wildcard characters that define patterns. They can be used with many programs, like AWK.

c	A single appearance of that character.
\c	If c is a special character, "escape" (not interpret) it. (\ \ "escapes" the very own \ character)
.	A single appearance of any character.
^	Stands for the start of the string or line.
\$	Stands for the end of the string or line.
[set]	Any of the characters inside that set would be valid.
[~set]	Any of the characters NOT inside that set would be valid.
	Either the character before and after would be valid.
?	The previous character can appear one or zero times.
+	The previous character can appear one or more times.
*	The previous character can appear zero or more times.

For command-line use, like with **ls**, **cp**, **rm** or **grep**:

\c	"Escape" a special character as seen above.
'string'	Any special character between single quotes is ignored.
"string"	Same as above, but won't ignore \$, \ or '.
[set]	Any of the characters inside that set would be valid.
?	A single appearance of any character.
*	Zero or more characters, no matter which ones.

Examples:

ls *.ogg	List any file that ends in <b>.ogg</b> .
cp ?.* /tmp	Copy any file named with one letter to <b>/tmp</b> .

## Searching

<code>grep PATTERN name</code>	Print any line of a file that has the pattern.
<code>grep -r PATTERN name</code>	Same as above, but seeking recursively inside all the files of a directory.
<code>name   grep PATTERN</code>	Search the output of a program.
<code>updatedb</code>	Create or update the database for <b>locate</b> .
<code>locate name</code>	List all files whose names contain <b>name</b> .
<code>find / -name "*img*"</code>	System-wide search for a file whose name contains "img" (expression-compatible).

## Filesystems

<code>fdisk -l</code>	List all devices and partitions.
<code>fdisk name</code>	Work in a device in interactive mode ( <b>M</b> shows help, <b>Q</b> quits).
<code>mkfs -t type name</code>	Create a filesystem in that partition.
<code>mkfs.ext4 name</code>	Same as above if EXT4 type was chosen (similar utilities for other types exist).
<code>mount name path</code>	Mount a device to that directory.
<code>mount -t type name path</code>	Same as above specifying type.
<code>umount name</code>	Unmount a mounted device.
<code>dd if=orig of=dest</code>	Copy / convert data, even partitions.

## Permissions

**Octal** mode (the order is user-group-others):

**4** is read, **2** is write, **1** is execute.

Examples:

<code>0700</code>	User can read, write and execute. The rest can't.
<code>0764</code>	Group can now read and write. Others can just read.
<code>0666</code>	Everyone can read and write but not execute.
<code>0025</code>	Group can write, others read and execute but not write.

**Symbolic** mode:

<b>u:</b>	User	<b>g:</b>	Group	<b>o:</b>	Others	<b>a:</b>	All (default)
<b>r:</b>	Read	<b>w:</b>	Write	<b>x:</b>	Execute		
<b>+</b> :	Add	<b>-:</b>	Remove	<b>=:</b>	Assign		

Examples:

<code>ug -w</code>	User and group are no longer able to write to it.
<code>go =rw</code>	Group and others can read and write.
<code>+x</code>	Make the file executable by everyone.

<code>chmod OCTAL name</code>	Change permissions of a file or directory.
<code>chmod SYMB name</code>	Same as above, using symbolic format.
<code>chmod OCTAL -R name</code>	Same as above, recursively (directories).
<code>umask OCTAL</code>	Set which permissions you DON'T want a file to have by default when created.
<code>chown user name</code>	Change the owner of a file.
<code>chown user:group name</code>	Same as above, specifying group.
<code>chown user -R name</code>	Change owner recursively (directories).

## Binaries and processes

<code>./name</code>	Execute that program in the current directory.
<code>./name &amp;</code>	Do it in the background (does not lock terminal).
<code>sh name.sh</code>	Execute a shell script.
<code>name1   name2</code>	Channel output of program <b>name1</b> as input for the program <b>name2</b> , using a system pipe.
<code>ps</code>	List all processes of this user on this terminal.
<code>ps aux</code>	Print full list of processes with details.
<code>top</code>	Dynamic, real-time list of processes. <b>Q</b> quits.
<code>kill pid</code>	Terminate the process with id <b>pid</b> .
<code>killall name</code>	Terminate any process named <b>name*</b> .
<code>nohup name</code>	Execute a command that won't terminate if the session ends (good for remote SSH).

## Services and modules

<code>service name status</code>	Show status for a given service.
<code>service name start</code>	Run a service calling the necessary scripts.
<code>service name stop</code>	Stop a service.
<code>service name restart</code>	Restart a service.
<code>modprobe name</code>	Load a .ko module into the kernel.
<code>modprobe -r name</code>	Remove a module from the kernel.

## Network

<code>ifconfig</code>	Show all network devices, and properties.
<code>ifconfig name up</code>	Enable a network device for use.
<code>ifconfig name down</code>	Disable a network device.
<code>dhclient</code>	Configure a DHCP connection.
<code>hostname</code>	Print the network name of the machine.
<code>hostname name</code>	Set the network name of the machine.
<code>ping host</code>	Check if a host is up with ICMP packets.
<code>traceroute host</code>	Show route (network hops) to a host.
<code>whois host</code>	Get WHOIS info of a host.
<code>dig host</code>	Get DNS info of a host.
<code>arp</code>	Print and edit the ARP cache.
<code>route</code>	Print and edit the routing table.
<code>ssh user@host</code>	Remote login as user <b>name</b> , encrypted.
<code>ssh user@host -P 600</code>	Same as above but using a given port.
<code>ssh user:pass@host</code>	Use that password instead of asking.
<code>scp name user@host:path</code>	Securely transfer a file using SSH.
<code>nc</code>	Run NetCat (TCP/IP swiss army knife).
<code>wget url</code>	Download a file or web page.
<code>wget -c url</code>	Same as above but allows pausing.

## Software

<code>which name</code>	Which binary will run if that order is issued.
<code>whereis name</code>	Locate binary, man page and other app's files.
<code>update-alternatives</code>	Set or show which programs will provide certain functionality, like <b>x-www-browser</b> .

### (Debian-based)

<code>apt-get update</code>	Update the list of available packages.
<code>apt-get upgrade</code>	Download and install all updates.
<code>apt-get search name</code>	Search a package by name.
<code>aptitude show name</code>	Show details of a software package.
<code>apt-get install name</code>	Install a software package.
<code>apt-get remove name</code>	Remove a software package (also <b>purge</b> ).
<code>dpkg -i name</code>	Install a .deb package.

## Development

<code>./configure</code>	Prepare a project for build using Autotools.
<code>make</code>	Build a Makefile-based project.
<code>make install</code>	Install a compiled project to the system.
<code>gcc files</code>	Compile one or more C source files (output will be a binary named <b>a.out</b> ).
<code>gcc files -o name</code>	Same as above, with a custom binary name.
<code>gcc files -I path</code>	Seek included headers in that directory.
<code>gcc files -L path</code>	Seek linked libraries in that directory.
<code>gcc files -l name</code>	Link a certain library to the binary.
<code>gcc files -static</code>	Use static linking.
<code>ldd name</code>	List the libraries needed by a binary.
<code>ldconfig</code>	Update the cache for the dynamic linker.

## X window system

<code>startx</code>	Start X session (GNOME, KDE...).
<code>CTRL+ALT+BACKSPACE</code>	Restart X server.
<code>service gdm stop</code>	Stop GNOME desktop manager.
<code>service kdm stop</code>	Stop KDE desktop manager.

## Bootloader

<code>grub-install name</code>	Install GRUB bootloader to a device.
<code>update-grub</code>	Update boot menu with all found OS.

## Important files

<code>/etc/default/grub</code>	GRUB settings (run update if changed).
<code>/etc/apt/sources.list</code>	Repository list for Apt (Debian-based).
<code>/etc/hostname</code>	Network name for this machine.
<code>/etc/resolv.conf</code>	List of DNS servers.
<code>/etc/hosts</code>	Known network hosts and their IPs.
<code>/etc/network/interfaces</code>	Network interfaces and configuration.
<code>/etc/fstab</code>	Storage devices and their mount options.
<code>~/.bashrc</code>	Shell script executed after every login.
<code>/etc/profile</code>	Same as above, but system-wide.