Linux Reference Page

Version 4, February 2014

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 **name**.
su - name	Same as above, but assuming the same environment that user would find.

Exiting

exit	Log out of a session (also **CTRL+D**).
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 **shutdown -h -P now**.
poweroff	Same as above.
reboot	Restart the computer.

Help, info, and output

man name	Show the manual page for that program. or important system file. **Q** quits.
apropos name	List man pages related to that **name**.
info name	More complete and updated info than **man**.
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 info about 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 file or 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 **name**.
deluser name	Remove the user with that **name**.
passwd name	Change the password of an user.
write name	Send messages to another user (**CTRL+D** quits).
mesg y	Enable the reception of messages via **write**.
mesg n	Disable the reception of messages via **write**.
mesg	Print the current state (if you receive messages).

Environment

printenv	Print all environment variables.
echo $NAME	Print the environment variable **NAME**.
export NAME=value	Create an environment variable.
env NAME=value name	Run a program in a custom environment.
alias name='command'	Create a shorcut 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 **name**.
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	Open a file with a console-based text editor.
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 back 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	Force the removal of the file (ignore warnings).
ln -s name1 name2	**name1** will be a symbolic link to file **name2**.
tar cf name files	Create a tar container with one or more files.
tar xf name	Extract the contents of a tar.
tar xfj name	Extract the contents of a bzip2 compressed tar.
tar xfz 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 **string** 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 **.ogg**.
cp ?? /tmp	Copy any file named with two characters to **/tmp**.

Searching

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

Filesystems

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

Permissions

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

4 is read, **2** is write, **1** is execute. You must add them. Examples:

0700	User can read, write and execute. The rest can't.
0764	Group can now read and write. Others can just read.
0025	Group can write, others read and execute but not write.

Symbolic mode:

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

Examples:

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

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

Binaries and processes

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

Services and modules

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

Network

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

Software

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

For Debian-based distributions:

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

Development

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

X window system

`startx`	Start X session (GNOME, KDE...).
`CTRL+ALT+BACKSPACE`	Restart X server.

Bootloader

`grub-install name`	Install GRUB bootloader to a device.
`update-grub`	Update boot menu with all found OS.

Important files

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