

# Linux Reference Page

Version 6, November 2024

[ ] used as placeholder

## Authentication

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 (root).
sudo [ ]	Run a command as superuser.
su [ ]	Log in as an user ( <b>su -</b> to assume the same environment that user would find).

## Exiting

exit	Log out of a session (also <b>CTRL+D</b> ).
systemctl poweroff	Turn the system off now, and don't reboot (append <b>-- HH:MM</b> to schedule it at a time).
halt	Same as <b>systemctl poweroff</b> .
poweroff	Same as above or <b>systemctl halt</b> .
reboot	Restart the computer.

## Help, info, and output

man [ ]	Show the manual page for a program or file.
apropos [ ]	List man pages related to a program or file.
info [ ]	More complete and updated info than <b>man</b> .
history	Show command log (!N re-executes number N).
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 [ ]	Show size info about a file or directory.
du -h [ ]	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 ( <b>/var/log/journal/*</b> ).

## Users

whoami	Print user name.
who	Info about active users.
who am i	Same as above but for current terminal user.
w	More detailed info about all active users.
users	Print all users logged in this machine.
useradd [ ]	Add a new user.
userdel [ ]	Remove an user.
passwd [ ]	Change the password of an user.
write [ ]	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 current <b>mesg</b> status ( <b>y</b> or <b>n</b> ).

## Environment

printenv	Print all environment variables.
echo \$[ ]	Print an environment variable.
export [VAR]=[VAL]	Create an environment variable.
env [VAR]=[VAL] [CMD]	Run a command in a custom environment.
alias [NAME]="[CMD]"	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:/path/to/directory
```

## Directories

pwd	Print current working directory.
cd	Change to your user \$HOME directory.
cd /path/to/dir	Change to a directory using its full path.
cd path/to/dir	Change to a directory using its relative path.
cd ..	Change to the parent directory.
cd -	Return to the previous directory.
mkdir [ ]	Create a new directory.
cp -r [1] [2]	Recursively copy a directory (1) to another (2).
mv [1] [2]	Move an empty directory (1) into another (2).
rmdir [ ]	Remove an empty directory.
rm -r [ ]	Remove a non-empty directory (recursive).
rm -rf [ ]	Force removal of a directory (ignore warnings).

## Files

In Linux, directories are considered files too, and extensions do not matter. 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 [ ]	Create a given file, or update its statistics.
nano [ ]	Open a file with a console-based text editor.
cat [ ]	Print the full contents of a file.
more [ ]	Same as above but interactive, for long files.
less [ ]	Same as above but able to go back and forth.
head -n N [ ]	Print the first N lines of a file (10 is default).
tail -n N [ ]	Print the last N lines of a file (10 is default).
tail -f [ ]	Keep printing as the file grows, good for logs.
cp [1] [2]	Copy a file (1) to a new destination/name (2).
mv [1] [2]	Same as above, but move (cut), not copy.
rm [ ]	Fully remove (skipping trash) a file.
rm -f [ ]	Force removal of a file (ignore warnings).
ln -s [NAME] [F]	Create a symbolic link to a file or directory.
tar cf [NAME] [F]	Create a tar container with one or more files.
tar xf [ ]	Extract the contents of a tar.
tar xjf [ ]	Extract the contents of a bzip2 compressed tar.
tar xzf [ ]	Extract the contents of a Gzip compressed tar.
[CMD] < [F]	Feed a file as input for a program.
[CMD] > [F]	Save output to a file, overwriting if exists.
[CMD] >> [F]	Same as above, but will append, not overwrite.
[CMD] << [STR]	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 “c” 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.
[abc]	Any of the characters inside the [ ] set would be valid.
[^abc]	Any of the characters NOT inside the set would be valid.
	Either the character before and after would be valid.
?	The previous character can appear once or not.
+	The previous character can appear one or more times.
*	The previous character can appear zero or more times.

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

\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 '.
[abc]	Any of the characters inside the [ ] set would be valid.
?	A single appearance of any character.
*	Zero or more characters, no matter which ones.

Examples:

ls *.txt	List any file whose name ends in “.txt”.
cp ?? /tmp	Copy any file named with two characters to /tmp.

## Searching

<code>grep [P] [F]</code>	Print any line of a file matching a pattern.
<code>grep -r [P] [D]</code>	Same as above, but seeking recursively inside all the files of a directory.
<code>[CMD1]   grep [P]</code>	Search the output of a program.
<code>updatedb</code>	Create or update the database for <b>locate</b> .
<code>locate []</code>	List all files (partially) matching a name.
<code>find / -name "*abc*"</code>	System-wide search for a file whose name contains “abc” (expression-compatible).

## Filesystems

<code>fdisk -l</code>	List all devices and partitions.
<code>parted []</code>	Work in a device in interactive mode ( <b>fdisk</b> is older but ubiquitous).
<code>fdisk []</code>	
<code>mkfs -t [T] [P]</code>	Format a partition with a filesystem.
<code>mkfs.ext4 [P]</code>	Same as above if EXT4 type was chosen (similar utilities for other types exist).
<code>mount [P] [D]</code>	Mount a partition to a directory.
<code>mount -t [T] [P] [D]</code>	Same as above specifying type.
<code>umount []</code>	Unmount a mounted partition or directory.
<code>dd if=[1] of=[2]</code>	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:

<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 [P] [F]</code>	Change permissions of a file or directory (either octal or symbolic syntax).
<code>chmod [P] -R [D]</code>	Same as above, recursively (directories).
<code>umask []</code>	Set which permissions you DON'T want a file to have by default when created.
<code>chown [USR] [F]</code>	Change the owner of a file or directory.
<code>chown [USR]:[GRP] [F]</code>	Same as above, specifying group.
<code>chown [USR] -R [D]</code>	Change owner recursively (directories).

## Binaries and processes

<code>./[]</code>	Execute a binary in the current directory.
<code>./[] &amp;</code>	Do it in the background (doesn't lock terminal).
<code>bash []</code>	Execute a .sh shell script file.
<code>[CMD1]   [CMD2]</code>	Channel output of the first command as input for the second, 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 []</code>	Terminate the process with that PID.
<code>killall abc</code>	Terminate any process matching “abc*”.
<code>CTRL+C</code>	Kill current process (can be intercepted).
<code>CTRL+Z</code>	Suspend current process (can be resumed later).
<code>nohup []</code>	Execute a command that won't terminate if the session ends (good for remote SSH).

## Services and modules

<code>systemctl status []</code>	Show status for a given service.
<code>systemctl start []</code>	Run a service calling the necessary scripts.
<code>systemctl stop []</code>	Stop a service.
<code>systemctl enable []</code>	Enable a service to run at boot.
<code>modprobe []</code>	Load a .ko module into the kernel.
<code>modprobe -r []</code>	Remove a module from the kernel.

## Network

<code>ifconfig -a</code>	Show all network devices and properties.
<code>ifconfig [] up</code>	Enable a network device for use.
<code>ifconfig [] down</code>	Disable a network device.
<code>dhclient []</code>	Try to negotiate a DHCP connection.
<code>hostname</code>	Print the network name of the machine.
<code>hostname []</code>	Set the network name of the machine.
<code>ping []</code>	Check if a host is up with ICMP packets.
<code>traceroute []</code>	Show route (network hops) to a host.
<code>whois []</code>	Get WHOIS info of a host.
<code>dig []</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 [USR]@[H]</code>	Secure remote login in a host, encrypted.
<code>ssh [USR]@[H] -P N</code>	Same as above but using port number N.
<code>ssh [USR]:[PASS]@[H]</code>	Use a password instead of asking.
<code>scp [1] [USR]@[H]:[2]</code>	Secure transfer a file (1) to destination (2).
<code>nc</code>	Run NetCat (TCP/IP swiss army knife).
<code>wget []</code>	Download a file or web page.
<code>wget -c []</code>	Same as above but allows pausing.

## Software

<code>which []</code>	Which binary will run if a command is issued.
<code>whereis []</code>	Locate binary, man page and other app's files.
<code>update-alternatives</code>	Set or show the default program to provide certain functionality, like “x-www-browser”.

For Debian-based distributions:

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

## Development

<code>./configure</code>	Prepare a project for build using Autotools.
<code>cmake ..</code>	Prepare a project for build using CMake (if CMakeLists.txt is in the parent folder).
<code>make -jN</code>	Build a Makefile project using N threads.
<code>make install</code>	Install a compiled project to the system.
<code>gcc []</code>	Compile one or more C source files (output will be a binary named “a.out”).
<code>gcc [F] -o [NAME]</code>	Same as above, with a custom binary name.
<code>gcc [F] -I [D]</code>	Seek included headers in a directory.
<code>gcc [F] -L [D]</code>	Seek linked libraries in a directory.
<code>gcc [F] -l [L]</code>	Link a certain library to the binary.
<code>gcc [F] -static</code>	Use static linking (embed libraries).
<code>ldd []</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.

## Bootloader

<code>grub-install []</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 partitions and mount options.
<code>~/bashrc</code>	Shell script executed after every login.
<code>/etc/profile</code>	Same as above, but system-wide.