Cheatography

Linux Commands & Options Cheat Sheet

by River L. (Tamaranth) via cheatography.com/42882/cs/46551/

Command Redirection

<	Gets input from somewhere other than stdin (keyboard)
>	Sends output somewhere other than stdout (monitor)
1>	Alternative to ">" (not really used)
>>	Redirects stdout; **appends** it to existing content (instead of overwriting it)
2>	Redirects only error messages (stderr)
&>	Redirects both stdout and stderr
cmd1 **|** *cmd2*	Pipes *cmd1* into *cmd2* (uses the **output** of cmd1 as the **input** for cmd2)

Basic File/Directory Commands

ls *dir1*	List contents of dir1
-l	additional info (owner, perms, etc.)
-a	lists all, including hidden files
-i	adds file inode to info displayed
-h	adds file size in human-readable format (e.g. MB instead of bytes)
-d	info for directory itself (rather than its contents)
mkdir *dir1*	Create a new directory
-p	creates full path (multiple directories if needed)
-v	verbose (output shows action taken)
cp *dir1 dir2*	Copy dir1 into dir2
-r	recursively ("and its contents")
rm *dir1*	Removes (deletes) an empty directory

Basic File/Directory Commands (cont)

-r	recursively ("and it's contents"), including special files
-i	interactively delete files (ask before deleting)
mv *file1 dir1*	Move file1 into dir1
-v	verbose (output shows action taken)
ln *target link*	Creates a hard link from file "link" to file "target"
-s	makes a symbolic link instead of a hard link

Locating Files

locate *pattern*	Search for file names with specified pattern (via "locate" database)
updatedb	Update the "locate" database
find */dir* **-***how what*	Recursive search starting at "*/dir*" for pattern ("what") of specified type ("how")
-name *name*	**case-sensitive** search for file name
-iname *name*	case-**in**sensitive search for file name
-user *name*	files owned by user with specified username
-uid *1002*	

By **River L.** (Tamaranth)
cheatography.com/tamaranth/

Not published yet.
Last updated 7th July, 2025.

Locating Files (cont)

-size *100M*	files that are **exactly** 100 MB in size
-size [**+**/**-**]*100M*	files that are [**larger**/**smaller**] than 100 MB
-type f	specifically files (not directories)
-perm /o=w	files where at least "**o**ther" has **w**rite perms
inum *inode*	file names associated with specified inode (i.e. including linked files)
-mmin *-5*	
-ctime	creation time

< `find /usr -size +100M` **`-exec ls -lh {} \;`** >
shows `ls -l` output AND actual file size of search results
(essentially "pipes" results of `find` into `ls -lh`)

Users & Groups

Command	*Options*	
id *user1*	Show UID, GID, & secondary groups (current user if not specified)	
getent passwd *usr1*	Find out if usr1 is known to the system	
useradd *user1*	Add a new user	
-u *#*	set a specific UID	
	-s /sbin/nologin	create a user without a login shell
usermod *user1*	Change properties of existing user	
-c "*text*"	adds text to comment field	

Users & Groups (cont)

-g *group1*	changes **primary group** to group1
-G *group1*	**replaces** supplementary group with group1
-aG *group1*	**appends** group1 to supplementary groups
-L *user1*	locks user1's account (instead of deleting)
newgrp *group1*	Change current user's primary group (**temporary**; current session only)
userdel *user1*	Deletes user1 but **not** their home directory
-r	deletes both user **and** home directory
Displays simple list of groups user1 is a member of	
passwd *user1*	Set password for user1
chage *user1*	Change password aging properties for user
-m *days*	minimum # of days between password changes
-M *days*	
-W *days*	warning period before password expires
-I *days*	inactivity period (password usable after expiration)
-d 0	require password change on next login

Users & Groups (cont)

-E *date*	date when **account** expires
getent group *grp1*	Find out if grp1 is known to the system
groupadd *group1*	Create a new group
-g *#*	set a specific GID
-r	create a **system** group
groupmod *group1*	Change properties of existing group
-g *#*	change GID to specified number

To set the number of days from today when user's **account** expires:
chage -E $(date +*days* **%Y-%m-%d)**

To give full admin privileges to a user or group:
echo "[*user1*|**%***group1*] **ALL=(ALL) ALL" >> /etc/s udo ers.d/***name*

Ownership & Permissions

chown *user*:*group*	Change owning user **and** owning group
chown :*group*	Change only owning group
chgrp *group file*	Alternate way to change owning group

chmod [*perms*] *file*	Change file/directory permissions
-R	recursive; applies to directory and all contents
g+r	
a-x	removes e**x**ecute perms for **a**ll parties (without changing other perms)

Ownership & Permissions (cont)

-R a=rwX	recursively set perms for **a**ll parties to **rw**, with **x** perms on directory **only**
770	octally set rwx perms for owning user & group, 0 for other
chmod [*perms*] *file*	(cont'd) Change special permissions
o+t *dir*	set **S**ticky bit (in **o**ther) * *For directories only** *
1*xxx*	
g+s *dir/file*	
2*xxx*	set SetGID bit – octal form
3*xxx*	
u+s *file*	
4*xxx*	
umask	Displays currently set umask (when used by itself)
077	removes default perms (0 from user, 7 from group/other)

By **River L.** (Tamaranth)
cheatography.com/tamaranth/

Not published yet.
Last updated 7th July, 2025.

Ownership & Permissions (cont)

`0077`	preserves any special perms when removing defaults

Octal Expression Values: **r=4** | **w=2** | **x=1**

Default permissions (on vanilla Linux):
directories=777 | **files=666**

By **River L.** (Tamaranth)
cheatography.com/tamaranth/

Not published yet.
Last updated 7th July, 2025.

Linux Commands & Options Cheat Sheet

by River L. (Tamaranth) via cheatography.com/42882/cs/46551/

Jobs & Load Handling

w	Displays currently logged-in users, login method, time, & resource usage info
--from	
-u *username*	
command **&**	Starts a new job in the background
jobs	Displays jobs running in the background ("**+**" = default job)
fg %*1*	Brings job 1 to the foreground
bg %*1*	Sends job 1 to the background
pidof *command*	Returns the PID(s) of a currently running job
systemctl status *cmd*	Shows main PID of a process, among other things (can use if `pidof` returns multiple)

--- CPU Load Handling ---

uptime	Displays the load average for the last 1, 5, and 15 mins
lscpu	Displays number of CPUs in the system, among other things

Jobs & Load Handling (cont)

-p	simplified output (each row of numbers = 1 CPU)

Process Priority Values: **-20** to **19** (default = **0**)
*(**negative** = **higher** priority | **positive** = **lower** priority)*

Viewing & Managing Processes

ps	Shows info about processes; pipe to `head`/`grep`/etc for less output (Note: "**[***output***]**" = kernel thread)
--forest	shows output in visual "tree" format

Shows processes in visual tree format
(may not always be installed)

-p

Linux terminal equivalent of Task Manager
(Also shows load average)
Keyboard Controls:

	k
	Shift+M
	Shift+P
	l / **t** / **m**
kill *PID*	Sends signal 15 to specified process ***** *Must use process ID* *****

By **River L.** (Tamaranth)
cheatography.com/tamaranth/

Not published yet.
Last updated 7th July, 2025.

Viewing & Managing Processes (cont)

-l	
killall *process*	Kills processes by name instead of PID
pkill	Look up & manage processes by name or other attributes (default is signal 15)
-*signal*	specify signal to send
-u *username*	sends signal to user's account (forces logout & shuts down their processes)
-t *pts/22*	kills specified terminal session (TTY)
nice -n *x command*	Starts a **new** process with an adjusted priority value (*x*)
renice -n *x* **-p** *PID*	Change the priority value (*x*) of an **existing** process

Managing Services & Daemons

--- System Commands ---	
systemctl reboot	reboots the system
systemctl daemon -reload	tells systemd you've made changes to a unit file & it needs to reinitialize those unit files
systemctl get-default	shows which target is the default target that the system summons whenever it starts up
systemctl set-default *graphical.target*	sets default target to "graphical.target"
--- Unit Commands ---	
systemctl list-units	shows all the active & loaded units on the system

Managing Services & Daemons (cont)

-t *type*	filters output by specified type of unit
-a	shows all units known to systemd (including "inactive" & "-not-found")
systemctl status *unit*	shows status & details of specified unit (including location of unit files)
systemctl is-enabled *unit.service*	shows whether the service is enabled (i.e. will start automatically on system boot)
systemctl is-active *unit.service*	shows whether the service is active
systemctl enable --now *unit.service*	enables the service & start it immediately (sets "enable" persistently)
systemctl disable --now *unit*	stops service immediately & sets to "disabled" persistently
systemctl reload *unit*	sends the "hang up" signal; drops config file loaded in memory & reload it from file system
systemctl restart *unit*	kills the process & starts it up again fresh (main PID will change)
systemctl reload -or -restart *unit*	reloads if possible, otherwise restart (for when you don't know if unit supports reload)
systemctl list-dependencies *unit.service*	shows list of units required by *unit.service* for it to work
--reverse	shows units that call *unit.service*
systemctl mask *unit*	prevents unit from being started automatically **or** manually (not even via `enable --now` command)
systemctl unmask *unit*	undoes masking on a service

Cheatography

Linux Commands & Options Cheat Sheet

by River L. (Tamaranth) via cheatography.com/42882/cs/46551/

File Systems & Block Devices

findmnt -s	
lsblk	Lists block devices in tree format (incl. size, mountpoint, etc.)
-f / **--fs**	
-p	shows full device paths
blkid *`/dev/file`*	Shows info about block device & filesystem, including partition UUID (non-partition devices must have a filesystem to show output)
df *`location`*	Shows **d**ata (storage) utilization of **f**ilesystem, device, & mountpoint (Shows info for **whole** filesystem, even if *`location`* is a subdirectory)
human-readable format (converts bytes to MB, GB, etc.)	
du *`/dir`*	Shows **d**ata (storage) **u**tilization of a specific directory
-s	

File Systems & Block Devices (cont)

-h	human-readable format (converts bytes to MB, GB, etc.)
mkfs.*`type /dev/file`*	Creates filesystem of specified type on specified block device (e.g. `mkfs.ext4 /dev/vdc1`)
mount *`/dev/file /dest`*	Takes filesystem on specified block device & mounts it in specified directory
umount *`mountpoint`*	Unmount the filesystem at the specified mountpoint
lsof *`mountpoint`*	**L**ist **o**pen **f**iles; shows which files are open in that filesystem
fuser *`mountpoint`*	Similar info as `lsof`, but less detailed
-m	shows only the PID associated with the open file

RPM Software Packages

rpm -qa	**Q**uery **a**ll; shows all software packages installed on the system
rpm -qf *`/file`*	**Q**uery **f**ile; asks RPM database which software package owns "*/file*"
rpm -qp	**Q**uery **p**ackage; runs command against the .rpm file itself

By **River L.** (Tamaranth)
cheatography.com/tamaranth/

Not published yet.
Last updated 7th July, 2025.

RPM Software Packages (cont)

`-qp`**`l`**	**l**ist of files in the .rpm package
`-qp`**`c`**	
`-qp`**`d`**	
`-qp`**`i`**	**i**nformation about the package file (metadata, summary, software description, etc.)

Managing Software Packages (DNF)

Info Commands:

`dnf info` *`package`*	Shows package metadata, summary, description, etc. (same as `rpm -qpi`, but package doesn't have to be installed)
`dnf provides` */file*	Shows what software package provides the file (same as `rpm -qf`, but package doesn't have to be installed)
`dnf search` '*gui*'	Searches for "*gui*" based on package name & metadata
`dnf search all` '*gui*'	Includes package description when searching for "*gui*"
`dnf list` '*x**'	Lists all available and/or installed software packages that begin with "*x*"

Managing Software Packages (DNF) (cont)

`dnf group list`	Shows list of available software groups ("Environment Groups" = logical groupings of regular groups from "-Available" list)
`dnf group info` "*Name*"	Lists software packages included in specified group
`dnf history`	Displays history of DNF commands/-actions
`dnf history undo` *3*	Undoes number *3* from DNF history (e.g. if it installed something, "undo" will uninstall it)

Download/Install Commands:

`dnf download --resolve` *`package`*	Downloads software package & its dependencies **without** installing them
`dnf install` *`package`*	Installs package & automatically resolves dependencies
`-y`	
`dnf remove` *`package`*	Uninstalls package & any dependencies (**if** they aren't being used elsewhere)
`-y`	auto-**y**es (for non-interactive)

Managing Software Packages (DNF) (cont)

dnf update *package*	Redownloads .rpm file & reinstalls newer version

-y

By **River L.** (Tamaranth)
cheatography.com/tamaranth/

Not published yet.
Last updated 7th July, 2025.

Cheatography

Linux Commands & Options Cheat Sheet

by River L. (Tamaranth) via cheatography.com/42882/cs/46551/

Network Info Commands

ip -br addr	
ip link show	Link-level properties of all interfaces (MAC addresses, etc.)
ip address show	Equivalent of `ipconfig` (same info as `link show`, plus IP addresses)
ip a s	
ip route show	Shows default route
ip r s	
ping | **ping6**	IPv4 ping | IPv6 ping
tracepath **tracepath6**	Equivalent of `tracert`/`traceroute` IPv6 version
mtr *address*	
nmap -sS *host*	Scans ports on host; shows port number/type, open state, associated service, etc.
ss	**S**ockets **s**tate information
-p	**p**rocesses responsible for opening ports
-l	
-u	

Network Info Commands (cont)

-n	translates names to **n**umbers (e.g. process name + PID)
-t	**T**CP sockets
nmcli connection show	Shows configured connection profiles
	nmcli c s abbreviated form
	--active only show currently active profiles
nmcli dev status	Status of interfaces (devices), incl. type & profile name
nmcli dev show *int*	Shows settings applied to the specified interface (device)

Network Configuration Commands (NMCLI)

nmcli con add <*specs*> *[options]*	Create a new connection profile **<specs>** = required configs *[options]* = other optional configs
<**con-name** "*Name*">	
<**type** *ethernet*>	
<**ifname** *eth0*>	interface to associate with profile
[**ipv4.method** ***manual***]	
[**ipv4.a ddr esses** *addres s/cidr*]	
[**ipv4.dns** *x.x.x.x*]	specify DNS server address
nmcli con mod "*Name*"	Modify properties of a connection profile

Network Configuration Commands (NMCLI) (cont)

ipv4.g ateway *x.x.x.x*	
+*property.attribute value*	
-*property.attribute value*	remove a value from an array of values
nmcli con up "*Name*"	Activates specified profile on whatever interface it's configured for
nmcli con reload "*Name*"	Reloads profile after config changes **Note:** Must re-**up** profile before new config will be applied to interface!

SSH

ssh *user@ip.addr*	Start ssh connection
-v	verbose; shows in detail what's happening while establishing connection
-Y	enables graphical application support
-p *port#*	connect to ssh service not listening on default port 22
-i *keyfile*	reads private key from identity file for public key authentication (not part of lecture; added from Google search when it came up in the quiz)

SSH (cont)

-o *option dest_server*	configures options when connecting to specified destination
Example: `PreferredAuthentications=password`	(Example: require a password; can use commas to specify multiple)
ssh-keygen	Generates a public/private key pair
-N '' -f *filename*	generates key pair without extra prompts (`''` = no passphrase; `-f` specifies non-default file location)
ssh-co py-id *user@ip*	Installs public key on destination server
-i *filepath*	

Misc. Commands

whatis *command*	Displays a short, one-line summary of what the command does
file *something*	Tells you what kind of file "*something*" is (More reliable than file extension)
echo	Creates output (`print` = Python equivalent)
date	Displaying, setting, calculating... basically all things date-related

Misc. Commands (cont)

+%F	display current date in international format (YYYY-mm-dd)
+%R	display current time (24-hr clock)
+%s	display # of seconds since epoch

hostname	Shows the hostname of the current system. That is all.
host *hostname*	Shows IP address associated with specified hostname
hostna mectl	Shows properties of current system (hostname, OS, kernel, etc.)
set-ho stname *new_name*	sets hostname to "*new_name*"

sos report	Generates an SOS report
-l	lists available plugins for SOS (recommended to pipe to `less`)
--upload	automatically associates generated report with your Red Hat account (enables browsing the data via the web portal)
sleep *seconds*	Does nothing for specified time (used for scripting)
tmux	Splits terminal window into multiple separate panes *(I think)*

Misc. Commands (cont)

gnome- cha racter s	Opens window of gnome icons/emotes
gnome- cal culato r	Opens built-in calculator (like Windows)

By **River L.** (Tamaranth)
cheatography.com/tamaranth/

Not published yet.
Last updated 7th July, 2025.