**The Essential Toolkit for the Terminal**

Linux includes a large number of commands, but we’ve chosen 37 of the most important ones to present here. Learn these commands, and you’ll be much more at home at the Linux command prompt.

The below list is presented in alphabetical order. A command’s position in the list is not representative of its usefulness or simplicity. For the final word on a command’s usage, refer to its man pages. The man command is in our list, of course—it’s short for “manual.”

**1. alias**

The alias command lets you give your own name to a command or sequence of commands. You can then type your short name, and the shell will execute the command or sequence of commands for you.

alias cls=clear

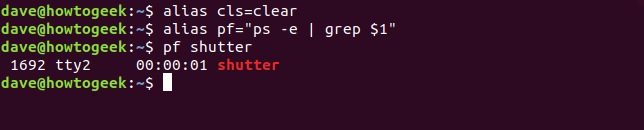
This sets up an alias called cls . It will be another name for clear . When you type cls, it will clear the screen just as though you had typed clear . Your alias saves a few keystrokes, sure. But, if you frequently move between Windows and Linux command line, you can find yourself typing the Windows cls command on a Linux machine that doesn’t know what you mean. Now it will know.

Aliases can be much more intricate than that simple example. Here’s an alias called pf (for process find) that is just a little more complex. Note the use of quotation marks around the command sequence. This is required if the command sequence has spaces in it. This alias uses the ps command to list the running processes and then [pipes](https://www.howtogeek.com/110150/become-a-linux-terminal-power-user-with-these-8-tricks/) them through the grep command. The grep command looks for entries in the output from ps that match the command line parameter $1 .

alias pf="ps -e | grep $1"

If you wanted to discover the process ID (PID) of the shutter process—or to find out if shutter was even running—you could use the alias like this. Type pf,  a space, and the name of the process you are interested in:

pf shutter

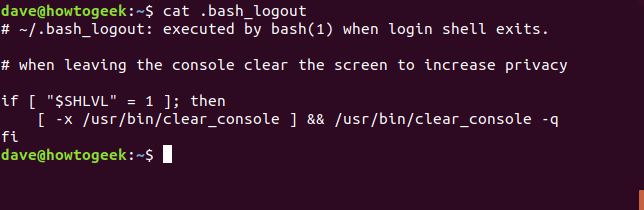


Aliases defined on the command line will die with the terminal window. When you close it, they are gone. To make your aliases always be available to you, add them to the.bash\_aliases file in your home directory.

**2. cat**

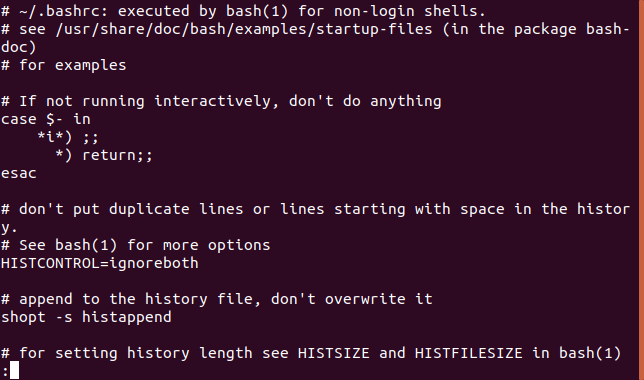
The cat command (short for “concatenate”) lists the contents of files to the terminal window. This is faster than opening the file in an editor, and there’s no chance you can accidentally alter the file. To read the contents of your .bash\_log\_out file, type the following command while the home directory is your current working directory, as it is by default:

cat .bash\_logout



With files longer than the number of lines in your terminal window, the text will whip past too fast for you to read. You can pipe the output from cat through less to make the process more manageable.  With less you can scroll forward and backward through the file using the Up and Down Arrow keys, the PgUp and PgDn keys, and the Home and End keys. Type q to quit from less.

cat .bashrc | less



**3. cd**

The cd command changes your current directory. In other words, it moves you to a new place in the filesystem.

If you are changing to a directory that is within your current directory, you can simply type cd and the name of the other directory.

cd work

If you are changing to a directory elsewhere within the filesystem directory tree, provide the path to the directory with a leading /.

cd /usr/local/bin

To quickly return to your home directory, use the ~ (tilde) character as the directory name.

cd ~

cd command in a terminal window

Here’s another trick: You can use the double dot symbol .. to represent the parent of the current directory. You can type the following command to go up a directory:

cd ..

Imagine you are in a directory. The parent directory has other directories in it, as well as the directory you’re currently in. To change into one of those other directories, you can use the .. symbol to shorten what you have to type.

cd ../games

cd command with .. in a terminal window

**4. chmod**

The chmod command [sets the file permissions flags](https://www.howtogeek.com/67987/htg-explains-how-do-linux-file-permissions-work/) on a file or folder. The flags define who can read, write to or execute the file. When you list files with the -l (long format) option you’ll see a string of characters that look like

-rwxrwxrwx

If the first character is a - the item is a file, if it is a d the item is a directory. The rest of the string is three sets of three characters. From the left, the first three represent the file permissions of the *owner*, the middle three represent the file permissions of the *group* and the rightmost three characters represent the permissions for *others*. In each set, an r stands for read, a w stands for write, and an x stands for execute.

If the r, w, or x character is present that file permission is granted. If the letter is not present and a - appears instead, that file permission is not granted.

One way to use chmod is to provide the permissions you wish to give to the owner, group, and others as a 3 digit number.  The leftmost digit represents the owner. The middle digit represents the group. The rightmost digit represents the others. The digits you can use and what they represent are listed here:

* **0:** No permission
* **1:** Execute permission
* **2:** Write permission
* **3:** Write and execute permissions
* **4:** Read permission
* **5:** Read and execute permissions
* **6:** Read and write permissions
* **7:** Read, write and execute permissions

Looking at our example.txt file, we can see that all three sets of characters are rwx. That means everyone has read, write and execute rights with the file.

To set the permission to be read, write, and execute (7 from our list) for the *owner;* read and write (6 from our list) for the *group;* and read and execute (5 from our list) for the *others* we’d need to use the digits 765 with the chmod command:

chmod -R 765 example.txt

chmod command in a terminal window

To set the permission to be read, write and execute (7 from our list) for the *owner*, and read and write (6 from our list) for the *group* and for the *others* we’d need to use the digits 766 with the chmod command:

chmod 766 example.txt

**5. chown**

The chown command allows you to change the owner and group owner of a file. Listing our example.txt file with ls -l we can see dave dave in the file description. The first of these indicates the name of the file owner, which in this case is the user dave. The second entry shows that the name of the group owner is also dave.  Each user has a default group created when the user is created. That user is the only member of that group. This shows that the file is not shared with any other groups of users.

You can use chown to change the owner or group, or both of a file. You must provide the name of the owner and the group, separated by a : character. You will need to use sudo. To retain dave as the owner of the file but to set mary as the group owner, use this command:

sudo chown dave:mary example.txt

chown command in a terminal window

To change both the owner and the group owner to mary, you would use the following command;

sudo chown mary:mary example.txt

To change the file so that dave is once more the file owner and the group owner, use this command:

sudo chown dave:dave example.txt

**6. curl**

The curl command is a tool to retrieve information and files from Uniform Resource Locators (URLs) or internet addresses.

The curl command may not be provided as a standard part of your Linux distribution. Use apt-get to install this package onto your system if you’re using Ubuntu or another Debian-based distribution. On other Linux distributions, use your Linux distribution’s package management tool instead.

sudo apt-get install curl

Suppose you want to retrieve a single file from a GitHub repository. There is no officially supported way to this. You’re forced to clone the entire repository. With curl however, we can retrieve the file we want on its own.

This command retrieves the file for us. Note that you need to specify the name of the file to save it in, using the -o (output) option. If you do not do this, the contents of the file are scrolled rapidly in the terminal window but not saved to your computer.

curl https://raw.githubusercontent.com/torvalds/linux/master/kernel/events/core.c -o core.c

https://www.howtogeek.com/pagespeed_static/1.JiBnMqyl6S.gif

If you don’t want to see the download progress information use the -s (silent) option.

curl -s https://raw.githubusercontent.com/torvalds/linux/master/kernel/events/core.c -o core.c

curl in a terminal window

**7. df**

The df command [shows the size, used space, and available space](https://www.howtogeek.com/409611/how-to-view-free-disk-space-and-disk-usage-from-the-linux-terminal) on the mounted filesystems of your computer.

Two of the most useful options are the -h (human readable) and -x (exclude) options. The human-readable option displays the sizes in Mb or Gb instead of in bytes. The exclude option allows you to tell df to discount filesystems you are not interested in. For example, the squashfs pseudo-filesystems that are created when you install an application with the snap command.

df -h -x squashfs

df command in a terminal window

**RELATED:** [***How to View Free Disk Space and Disk Usage From the Linux Terminal***](https://www.howtogeek.com/409611/how-to-view-free-disk-space-and-disk-usage-from-the-linux-terminal/)

**8. diff**

The diff command [compares two text files](https://www.howtogeek.com/410532/how-to-compare-two-text-files-in-the-linux-terminal/) and shows the differences between them. There are many options to tailor the display to your requirements.

The -y (side by side) option shows the line differences side by side. The -w (width) option lets you specify the maximum line width to use to avoid wraparound lines. The two files are called alpha1.txt and alpha2.txt in this example. The --suppress-common-lines prevents diff from listing the matching lines, letting you focus on the lines which have differences.

diff -y -W 70 alpha1.txt alpha2.txt --suppress-common-lines

diff command in a terminal window

**RELATED:** [***How to Compare Two Text Files in the Linux Terminal***](https://www.howtogeek.com/410532/how-to-compare-two-text-files-in-the-linux-terminal/)

**9. echo**

The echo command prints (echoes) a string of text to the terminal window.

The command below will print the words “A string of text” on the terminal window.

echo A string of text

The echo command can show the value of environment variables, for example, the $USER, $HOME, and $PATH environment variables. These hold the values of the name of the user, the user’s home directory, and the path searched for matching commands when the user types something on the command line.

echo $USER

echo $HOME

echo $PATH

echo command in a terminal window

The following command will cause a bleep to be issued. The -e (escape code) option interprets the escaped a character as [a ‘bell’ character](https://en.wikipedia.org/wiki/Bell_character).

echo -e "\a"

The echo command is also invaluable in shell scripts. A script can use this command to generate visible output to indicate the progress or results of the script as it is executed.

**10. exit**

The exit command will close a terminal window, end the execution of a shell script, or log you out of an SSH remote access session.

exit

exit command in a terminal window

**11. find**

Use the find command to track down files that you know exist if you can’t remember where you put them. You must tell find where to start searching from and what it is looking for. In this example, the . matches the current folder and the -name option tells find to look for files with a name that matches the search pattern.

You can use wildcards, where \* represents any sequence of characters and ? represents any single character. We’re using \*ones\* to match any file name containing the sequence “ones.” This would match words like bones, stones, and lonesome.

find . -name \*ones\*

find command in a terminal window

As we can see, find has returned a list of matches. One of them is a directory called Ramones. We can tell find to restrict the search to files only. We do this using the -type option with the f parameter. The f parameter stands for files.

find . -type f -name \*ones\*

If you want the search to be case insensitive use the -iname (insensitive name) option.

find . -iname \*wild\*

**12. finger**

The finger command gives you a short dump of information about a user, including the time of the user’s last login, the user’s home directory, and the user account’s full name.

finger command in a terminal window

**13. free**

The free command gives you a summary of the memory usage with your computer. It does this for both the main Random Access Memory (RAM) and swap memory. The -h (human) option is used to provide human-friendly numbers and units. Without this option, the figures are presented in bytes.

free -h

free command in a terminal window

**14. grep**

The grep utility searches for lines which contain a search pattern. When we looked at the alias command, we used grep to search through the output of another program, ps . The grep command can also search the contents of files. Here we’re searching for the word “train” in all text files in the current directory.

grep train \*.txt

The output lists the name of the file and shows the lines that match. The matching text is highlighted.

grep command in a terminal window

The functionality and sheer usefulness of grep definitely warrants you checking out [its man page](http://man7.org/linux/man-pages/man1/grep.1.html).

**15. groups**

The groups command tells you which groups a user is a member of.

groups dave

groups mary

groups command in a terminal window

**16. gzip**

The gzip command compresses files. By default, it removes the original file and leaves you with the compressed version. To retain both the original and the compressed version, use the -k (keep) option.

gzip -k core.c

gzip command in a terminal window

**17. head**

The head command gives you a listing of the first 10 lines of a file. If you want to see fewer or more lines, use the -n (number) option. In this example, we use head with its default of 10 lines. We then repeat the command asking for only five lines.

head -core.c

head -n 5 core.c

head command in a terminal window

**18. history**

The history command lists the commands you have previously issued on the command line. You can repeat any of the commands from your history by typing an exclamation point ! and the number of the command from the history list.

!188

history command in a terminal window

Typing two exclamation points repeats your previous command.

!!

**19. kill**

The kill command allows you to terminate a process from the command line. You do this by providing the process ID (PID) of the process to kill. Don’t kill processes willy-nilly. You need to have a good reason to do so. In this example, we’ll pretend the shutter program has locked up.

To find the PID of shutter we’ll use our ps and grep trick from the section about the alias command, above. We can search for the shutter process and obtain its PID as follows:

ps -e | grep shutter.

Once we have determined the PID—1692 in this case—we can kill it as follows:

kill 1692

kill command in a terminal window

**20. less**

The less command allows you to view files without opening an editor. It’s faster to use, and there’s no chance of you inadvertently modifying the file. With less you can scroll forward and backward through the file using the Up and Down Arrow keys, the PgUp and PgDn keys and the Home and End keys. Press the Q key to quit from less.

To view a file provide its name to less as follows:

less core.c

less command in a terminal window

You can also pipe the output from other commands into less. To see the output from ls for a listing of your entire hard drive, use the following command:

ls -R / | less

less command in a terminal window

Use / to search forward in the file and use ? to search backward.

**21. ls**

This might be the first command the majority of Linux users meet. It lists the files and folders in the directory you specify. By default, ls looks in the current directory. There are a great many options you can use with ls , and we strongly advise reviewing its [the man page](http://man7.org/linux/man-pages/man1/ls.1.html). Some common examples are presented here.

To list the files and folders in the current directory:

ls

To list the files and folders in the current directory with a detailed listing use the -l (long) option:

ls -l

To use human-friendly file sizes include the -h (human) option:

ls -lh

To include hidden files use the -a (all files) option:

ls -lha

ls command in a terminal window

**22. man**

The man command displays the “man pages” for a command in less . The man pages are the user manual for that command. Because man uses less to display the man pages, you can use the search capabilities of less.

For example, to see the man pages for chown, use the following command:

man chown

Use the Up and Down arrow or PgUp and PgDn keys to scroll through the document. Press q to quit the man page or pressh for help.

man command in a terminal window

**23. mkdir**

The mkdir command allows you to create new directories in the filesystem. You must provide the name of the new directory to mkdir. If the new directory is not going to be within the current directory, you must provide the path to the new directory.

To create two new directories in the current directory called “invoices” and “quotes,” use these two commands:

mkdir invoices

mkdir quotes

mkdir command in a terminal window

To create a new directory called “2019” inside the “invoices” directory, use this command:

mkdir invoices/2109

If you are going to create a directory, but its parent directory does not exist, you can use the -p (parents) option to have mkdir create all of the required parent directories too. In the following command, we are creating the “2019” directory inside the “yearly” directory inside the “quotes” directory. The “yearly” directory does not exist, but we can have mkdir create all the specified directories at once:

mkdir -p quotes/yearly/2019

The “yearly” directory is also created.

**24. mv**

The mv command allows you to move files and directories from directory to directory. It also allows you to rename files.

To move a file you must tell mv where the file is and where you want it to be moved to. In this example, we’re moving a file called apache.pdf from the “~/Document/Ukulele” directory and placing it in the current directory, represented by the single . character.

mv ~/Documents/Ukulele/Apache.pdf .

**mv command in a terminal window**

To rename the file, you “move” it into a new file with the new name.

mv Apache.pdf The\_Shadows\_Apache.pdf

The file move and rename action could have been achieved in one step:

mv ~/Documents/Ukulele/Apache.pdf ./The\_Shadows\_Apache.pdf

**25. passwd**

The passwd command lets you change the password for a user. Just type passwd to change your own password.

You can also change the password of another user account, but you must use sudo. You will be asked to enter the new password twice.

sudo passwd mary

passwd command in a terminal window

**26. ping**

The ping command lets you verify that you have network connectivity with another network device. It is commonly used to help troubleshoot networking issues. To use ping, provide the IP address or machine name of the other device.

ping 192.168.4.18

The ping command will run until you stop it with Ctrl+C.

ping in a terminal window

Here’s what’s going on here:

* The device at IP address 192.168.4.18 is responding to our ping requests and is sending back packets of 64 bytes.
* The [Internet Control Messaging Protocol](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) (ICMP) sequence numbering allows us to check for missed responses (dropped packets).
* The TTL figure is the “time to live” for a packet. Each time the packet goes through a router, it is (supposed to be) decremented by one. If it reaches zero the packet is thrown away. The aim of this is to prevent network loopback problems from flooding the network.
* The time value is the duration of the round trip from your computer to the device and back. Simply put, the lower this time, the better.

To ask ping to run for a specific number of ping attempts, use the -c (count) option.

ping -c 5 192.168.4.18

To hear a ping, use the -a (audible) option.

ping -a 192.168.4.18

**27. ps**

The ps command lists running processes. Using ps without any options causes it to list the processes running in the current shell.

ps

ps command in a terminal window

To see all the processes related to a particular user, use the -u (user) option. This is likely to be a long list, so for convenience pipe it through less.

ps -u dave | less

ps command in a terminal window

To see every process that is running, use the -e (every process) option:

ps -e | less

**28. pwd**

Nice and simple, the pwd command prints the working directory (the current directory) from the root / directory.

pwd

pwd command in a terminal window

**29. shutdown**

The shutdown command lets you [shut down or reboot your Linux system](https://www.howtogeek.com/411925/how-to-reboot-or-shut-down-linux-using-the-command-line/).

Using shutdown with no parameters will shut down your computer in one minute.

shutdown

shutdown command in a terminal window

To shut down immediately, use the now parameter.

shutdown now

shutdown now

You can also schedule a shutdown and inform any logged in users of the pending shutdown. To let the shutdown command know when you want it to shut down, you provide it with a time. This can be a set number of minutes from now, such as +90 or a precise time, like 23:00. Any text message you provide is broadcast to logged in users.

shutdown 23:00 Shutdown tonight at 23:00, save your work and log out before then!

shutdown 23:00 with message

To cancel a shutdown, use the -c (cancel) option. Here we have scheduled a shutdown for fifteen minutes time from now—and then changed our minds.

shutdown +15 Shutting down in 15 minutes!

shutdown -c

Shutdown -c cancel command

**RELATED:** [***How to Reboot or Shut Down Linux Using the Command Line***](https://www.howtogeek.com/411925/how-to-reboot-or-shut-down-linux-using-the-command-line/)

**30. SSH**

Use the ssh command to make a connection to a remote Linux computer and log into your account. To make a connection, you must provide your user name and the IP address or domain name of the remote computer. In this example, the user mary is logging into the computer at 192.168.4.23. Once the connection is established, she is asked for her password.

ssh mary@192.168.4.23

ssh command in a terminal window

Her user name and password are verified and accepted, and she is logged in. Notice that her prompt has changed from “Nostromo” to “howtogeek.”

Mary issues the w command to list the current users on “howtogeek” system. She is listed as being connected from pts/1, which is a pseudo-terminal slave. That is, it is not a terminal directly connected to the computer.

To close the session, mary types exit and is returned to the shell on the “Nostromo” computer.

w

exit

w and exit commands in a terminal window

**31. sudo**

The sudo command is required when performing actions that require root or superuser permissions, such as changing the password for another user.

sudo passwd mary

passwd command in a terminal window

**32. tail**

The tail command gives you a listing of the last 10 lines of a file. If you want to see fewer or more lines, use the -n (number) option. In this example, we use tail with its default of 10 lines. We then repeat the command asking for only five lines.

tail core.c

tail -n 5 core.c

tail command in a terminal window

**33. tar**

With the tar command, you can create an archive file (also called a tarball) that can contain many other files. This makes it much more convenient to distribute a collection of files. You can also use tar to extract the files from an archive file. It is common to ask tar to compress the archive. If you do not ask for compression, the archive file is created uncompressed.

To create an archive file, you need to tell tar which files to include in the archive file, and the name you wish the archive file to have.

In this example, the user is going to archive all of the files in the Ukulele directory, which is in the current directory.

ls command in the terminal window

They have used the -c (create) option and the -v (verbose) option. The verbose option gives some visual feedback by listing the files to the terminal window as they are added to the archive. The -f (filename) option is followed by the desired name of the archive. In this case, it is songs.tar.

tar -cvf songs.tar Ukulele/

tar -cvf command in a terminal window

The files are listed to the terminal window as they are added to the archive file.

There are two ways to tell tar that you want the archive file to be compressed. The first is with the -z (gzip) option. This tells tar to use the gzip utility to compress the archive once it has been created.

It is usual to add “.gz” as suffix to this type of archive. That allows anyone who is extracting files from it to know which commands to pass to tar to correctly retrieve the files.

tar -cvzf songs.tar.gz Ukulele/

tar -cvzf command in a terminal window

The files are listed to the terminal window as they are added to the archive file as before, but the creation of the archive will take a little longer because of the time required for the compression.

To create an archive file that is compressed using a superior compression algorithm giving a smaller archive file use the -j (bzip2) option.

tar -cvjf songs.tar.bz2 Ukulele/

tar -cvjf command in a terminal window

Once again, the files are listed as the archive is created. The -j option is noticeably slower than the -z option.

If you are archiving a great many files, you must choose between the -z option for decent compression and reasonable speed, or the -j option for better compression and slower speed.

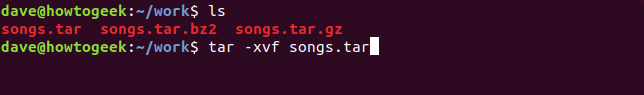
As can be seen in the screenshot below, the “.tar” file is the largest, the “.tar.gz” is smaller, and the “.tar.bz2” is the smallest of the archives.

ls command in a terminal window

To extract files from an archive file use the -x (extract) option. The -v (verbose) and -f (filename) options behave as they do when creating archives. Use ls to confirm which type of archive you are going to extract the files from, then issue the following command.

ls

tar -xvf songs.tar



The files are listed as they are extracted. Note that the Ukulele directory is also recreated for you.

To extract files from a “.tar.gz” archive, use the -z (gzip) option.

tar -xvzf songs.tar.gz

tar -xvzf command in a terminal window

Finally, to extract files from a “.tar.bz2” archive use the -j option instead of the -z (gzip) option.

tar -xvjf songs.tar.bz2

tar -xvjf command in a terminal window

**RELATED:** [***How to Extract Files From a .tar.gz or .tar.bz2 File on Linux***](https://www.howtogeek.com/409742/how-to-extract-files-from-a-.tar.gz-or-.tar.bz2-file-on-linux/)

**34. top**

The top command shows you a real-time display of the data relating to your Linux machine. The top of the screen is a status summary.

The first line shows you the time and how long your computer has been running for, how many users are logged into it, and what the load average has been over the past one, five, and fifteen minutes.

The second line shows the number of tasks and their states: running, stopped, sleeping and zombie.

The third line shows CPU information. Here’s what the fields mean:

* us: value is the CPU time the CPU spends executing processes for users, in “user space”
* sy: value is the CPU time spent on running system “kernel space” processes
* ni: value is the CPU time spent on executing processes with a manually set nice value
* id: is the amount of CPU idle time
* wa: value is the time the CPU spends waiting for I/O to complete
* hi: The CPU time spent servicing hardware interrupts
* si: The CPU time spent servicing software interrupts
* st: The CPU time lost due to running virtual machines (“steal time”)

The fourth line shows the total amount of physical memory, and how much is free, used and buffered or cached.

The fifth line shows the total amount of swap memory, and how much is free, used and available  (taking into account memory that is expected to be recoverable from caches).

top command in a terminal window

The user has pressed the E key to change the display into more humanly digestible figures instead of long integers representing bytes.

The columns in the main display are made up of:

* PID: Process ID
* USER: Name of the owner of the process
* PR: [Process priority](https://www.howtogeek.com/411979/how-to-set-process-priorities-with-the-nice-and-renice-commands-in-linux/)
* NI: The nice value of the process
* VIRT: Virtual memory used by the process
* RES: Resident memory used by the process
* SHR: Shared memory used by the process
* S: Status of the process. See the list below of the values this field can take
* %CPU: the share of CPU time used by the process since last update
* %MEM: share of physical memory used
* TIME+: total CPU time used by the task in hundredths of a second
* COMMAND: command name or command line (name + options)

(The command column didn’t fit into the screenshot.)

The status of the process can be one of:

* D: Uninterruptible sleep
* R: Running
* S: Sleeping
* T: Traced (stopped)
* Z: Zombie

Press the Q key to exit from top.

**RELATED:** [***How to Set Process Priorities With nice and renice on Linux***](https://www.howtogeek.com/411979/how-to-set-process-priorities-with-the-nice-and-renice-commands-in-linux/)

**35. uname**

You can obtain some system information regarding the Linux computer you’re working on with the uname command.

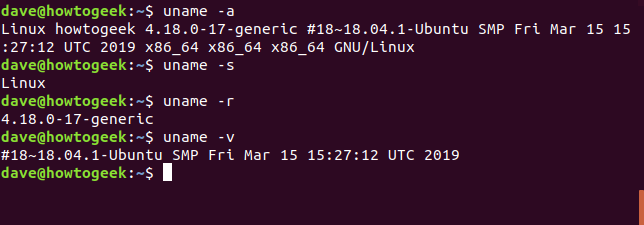
* Use the -a (all) option to see everything.
* Use the -s (kernel name) option to see the type of kernel.
* Use the -r (kernel release) option to see the kernel release.
* Use the -v (kernel version) option to see the kernel version.

uname -a

uname -s

uname -r

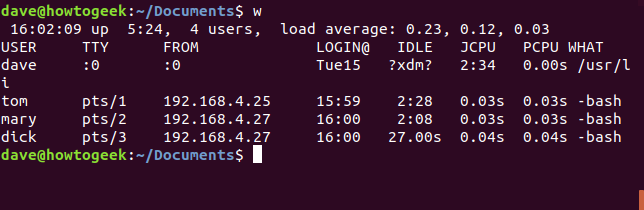
uname -v



**36. w**

The w command lists the currently logged in users.

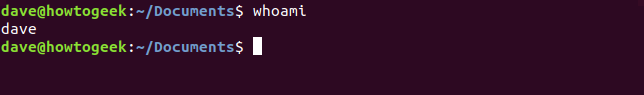
w



**37. whoami**

Use whoami to find out who you are logged in as or who is logged into an unmanned Linux terminal.

whoami



**RELATED:** [***How to Determine the Current User Account in Linux***](https://www.howtogeek.com/410423/how-to-determine-the-current-user-account-in-linux/)