**Uptime Command**

In Linux **uptime** command shows since how long your system is running and the number of users are currently logged in and also displays load average for **1,5** and **15** minutes intervals.

**# uptime**

08:16:26 up 22 min, 1 user, load average: 0.00, 0.03, 0.22

**Check Uptime Version**

**Uptime** command don’t have other options other than **uptime** and **version**. It gives information only in **hours:mins** if it less than **1** day.

[tecmint@tecmint ~]$ uptime -V

procps version 3.2.8

**2. W Command**

It will displays users currently logged in and their process along-with shows **load averages**. also shows the **login name**, **tty name**, **remote host**, **login time**, **idle time**, **JCPU**, **PCPU**, command and processes.

**# w**

08:27:44 up 34 min, 1 user, load average: 0.00, 0.00, 0.08

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

tecmint pts/0 192.168.50.1 07:59 0.00s 0.29s 0.09s w

**Available options**

1. **-h** : displays no header entries.
2. **-s** : without JCPU and PCPU.
3. **-f** : Removes from field.
4. **-V** : (upper letter) – Shows versions.

**3. Users Command**

Users command displays currently logged in users. This command don’t have other parameters other than help and version.

**# users**

tecmint

**4. Who Command**

**who** command simply return **user name**, **date**, **time** and **host information**. who command is similar to **w** command. Unlike **w** command **who** doesn’t print what users are doing. Lets illustrate and see the different between **who** and **w** commands.

**# who**

tecmint pts/0 2012-09-18 07:59 (192.168.50.1)

**# w**

08:43:58 up 50 min, 1 user, load average: 0.64, 0.18, 0.06

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

tecmint pts/0 192.168.50.1 07:59 0.00s 0.43s 0.10s w

**Who command Options**

1. **-b** : Displays last system reboot date and time.
2. **-r** : Shows current runlet.
3. **-a, –all** : Displays all information in cumulatively.

**5. Whoami Command**

**whoami** command print the name of current user. You can also use “**who am i**” command to display the current user. If you are logged in as a root using sudo command “**whoami**” command return **root** as current user. Use “**who am i**” command if you want to know the exact user logged in.

**# whoami**

tecmint

**6. ls Command**

ls command display list of files in human readable format.

**# ls -l**

total 114

dr-xr-xr-x. 2 root root 4096 Sep 18 08:46 bin

dr-xr-xr-x. 5 root root 1024 Sep 8 15:49 boot

Sort file as per last modified time.

**# ls -ltr**

total 40

-rw-r--r--. 1 root root 6546 Sep 17 18:42 install.log.syslog

-rw-r--r--. 1 root root 22435 Sep 17 18:45 install.log

-rw-------. 1 root root 1003 Sep 17 18:45 anaconda-ks.cfg

For more examples of ls command, please check out our article on [15 Basic ‘ls’ Command Examples in Linux](http://www.tecmint.com/15-basic-ls-command-examples-in-linux/).

**7. Crontab Command**

List schedule jobs for current user with **crontab** command and **-l** option.

**# crontab -l**

00 10 \* \* \* /bin/ls >/ls.txt

Edit your **crontab** with **-e** option. In the below example will open schedule jobs in **VI editor**. Make a necessary changes and quit pressing **:wq** keys which saves the setting automatically.

**# crontab -e**

For more examples of **Linux Cron Command**, please read our earlier article on [11 Cron Scheduling Task Examples in Linux](http://www.tecmint.com/11-cron-scheduling-task-examples-in-linux/).

**8. Less Command**

**less** command allows quickly view file. You can page up and down. Press ‘**q**‘ to quit from less window.

**# less install.log**

Installing setup-2.8.14-10.el6.noarch

warning: setup-2.8.14-10.el6.noarch: Header V3 RSA/SHA256 Signature, key ID c105b9de: NOKEY

Installing filesystem-2.4.30-2.1.el6.i686

Installing ca-certificates-2010.63-3.el6.noarch

Installing xml-common-0.6.3-32.el6.noarch

Installing tzdata-2010l-1.el6.noarch

Installing iso-codes-3.16-2.el6.noarch

**9. More Command**

**more** command allows quickly view file and shows details in percentage. You can page up and down. Press ‘**q**‘ to quit out from more window.

**# more install.log**

Installing setup-2.8.14-10.el6.noarch

warning: setup-2.8.14-10.el6.noarch: Header V3 RSA/SHA256 Signature, key ID c105b9de: NOKEY

Installing filesystem-2.4.30-2.1.el6.i686

Installing ca-certificates-2010.63-3.el6.noarch

Installing xml-common-0.6.3-32.el6.noarch

Installing tzdata-2010l-1.el6.noarch

Installing iso-codes-3.16-2.el6.noarch

**--More--(10%)**

**10. CP Command**

Copy file from source to destination preserving same mode.

**# cp -p fileA fileB**

You will be prompted before overwrite to file.

**# cp -i fileA fileB**

**11. MV Command**

Rename **fileA** to **fileB**. **-i** options prompt before overwrite. Ask for confirmation if exist already.

**# mv -i fileA fileB**

**12. Cat Command**

**cat** command used to view multiple file at the same time.

**# cat fileA fileB**

You combine **more** and **less** command with cat command to view file contain if that doesn’t fit in single screen / page.

**# cat install.log | less**

**# cat install.log | more**

For more examples of Linux cat command read our article on [13 Basic Cat Command Examples in Linux](http://www.tecmint.com/13-basic-cat-command-examples-in-linux/).

**13. Cd command (change directory)**

with cd command (change directory) it will goes to **fileA** directory.

**# cd /fileA**

**14. pwd command (print working directory)**

**pwd** command return with present working directory.

**# pwd**

/root

**15. Sort command**

Sorting lines of text files in ascending order. with **-r** options will sort in descending order.

**#sort fileA.txt**

**#sort -r fileA.txt**

**16. VI Command**

**Vi** is a most popular text editor available most of the **UNIX-like OS**. Below examples open file in read only with **-R** option. Press ‘**:q**‘ to quit from vi window.

**# vi -R /etc/shadows**

**17. SSH Command (Secure Shell)**

SSH command is used to login into remote host. For example the below ssh command will connect to remote host (**192.168.50.2**) using user as **narad**.

**# ssh narad@192.168.50.2**

To check the version of ssh use option **-V** (uppercase) shows version of ssh.

**# ssh -V**

OpenSSH\_5.3p1, OpenSSL 1.0.0-fips 29 Mar 2010

**18. Ftp or sftp Command**

**ftp** or **sftp** command is used to connect to remote ftp host. ftp is (**file transfer protocol**) and sftp is (**secure file transfer protocol**). For example the below commands will connect to ftp host (**192.168.50.2**).

**# ftp 192.168.50.2**

**# sftp 192.168.50.2**

Putting multiple files in remote host with **mput** similarly we can do **mget** to download multiple files from remote host.

**# ftp > mput \*.txt**

**# ftp > mget \*.txt**

**19. Service Command**

**Service** command call script located at **/etc/init.d/** directory and execute the script. There are two ways to start the any service. For example we start the service called **httpd** with service command.

**# service httpd start**

OR

**# /etc/init.d/httpd start**

**20. Free command**

**Free** command shows **free**, **total** and **swap memory** information in bytes.

**# free**

total used free shared buffers cached

Mem: 1030800 735944 294856 0 51648 547696

-/+ buffers/cache: 136600 894200

Swap: 2064376 0 2064376

Free with **-t** options shows **total memory** used and available to use in bytes.

**# free -t**

total used free shared buffers cached

Mem: 1030800 736096 294704 0 51720 547704

-/+ buffers/cache: 136672 894128

Swap: 2064376 0 2064376

Total: 3095176 736096 2359080

**21. Top Command**

**top** command displays processor activity of your system and also displays tasks managed by kernel in real-time. It’ll show **processor** and **memory** are being used. Use top command with **‘u**‘ option this will display specific User process details as shown below. Press ‘**O**‘ (**uppercase letter**) to sort as per desired by you. Press ‘**q**‘ to quit from top screen.

**# top -u tecmint**

top - 11:13:11 up 3:19, 2 users, load average: 0.00, 0.00, 0.00

Tasks: 116 total, 1 running, 115 sleeping, 0 stopped, 0 zombie

Cpu(s): 0.0%us, 0.3%sy, 0.0%ni, 99.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 1030800k total, 736188k used, 294612k free, 51760k buffers

Swap: 2064376k total, 0k used, 2064376k free, 547704k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

1889 tecmint 20 0 11468 1648 920 S 0.0 0.2 0:00.59 sshd

1890 tecmint 20 0 5124 1668 1416 S 0.0 0.2 0:00.44 bash

6698 tecmint 20 0 11600 1668 924 S 0.0 0.2 0:01.19 sshd

6699 tecmint 20 0 5124 1596 1352 S 0.0 0.2 0:00.11 bash

For more about top command we’ve already compiled a list of [12 TOP Command Examples in Linux](http://www.tecmint.com/12-top-command-examples-in-linux/).

**22. Tar Command**

**tar** command is used to compress files and folders in Linux. For example the below command will create a archive for **/home** directory with file name as **archive-name.tar**.

**# tar -cvf archive-name.tar /home**

To extract tar archive file use the option as follows.

**# tar -xvf archive-name.tar**

To understand more about **tar command** we’ve created a complete **how-to guide** on tar command at [18 Tar Command Examples in Linux](http://www.tecmint.com/18-tar-command-examples-in-linux/).

**23. Grep Command**

grep search for a given string in a file. Only **tecmint** user displays from **/etc/passwd** file. we can use **-i** option for ignoring case sensitive.

**# grep tecmint /etc/passwd**

tecmint:x:500:500::/home/tecmint:/bin/bash

**24. Find Command**

Find command used to search **files**, **strings** and **directories**. The below example of find command search **tecmint** word in ‘**/**‘ partition and return the output.

**# find / -name tecmint**

/var/spool/mail/tecmint

/home/tecmint

/root/home/tecmint

For complete guide on **Linux find command** examples fount at [35 Practical Examples of Linux Find Command](http://www.tecmint.com/35-practical-examples-of-linux-find-command/).

**25. lsof Command**

**lsof** mean List of all open files. Below lsof command list of all opened files by user **tecmint**.

**# lsof -u tecmint**

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

sshd 1889 tecmint cwd DIR 253,0 4096 2 /

sshd 1889 tecmint txt REG 253,0 532336 298069 /usr/sbin/sshd

sshd 1889 tecmint DEL REG 253,0 412940 /lib/libcom\_err.so.2.1

sshd 1889 tecmint DEL REG 253,0 393156 /lib/ld-2.12.so

sshd 1889 tecmint DEL REG 253,0 298643 /usr/lib/libcrypto.so.1.0.0

sshd 1889 tecmint DEL REG 253,0 393173 /lib/libnsl-2.12.so

sshd 1889 tecmint DEL REG 253,0 412937 /lib/libkrb5support.so.0.1

sshd 1889 tecmint DEL REG 253,0 412961 /lib/libplc4.so

For more **lsof command examples** visit [10 lsof Command Examples in Linux](http://www.tecmint.com/10-lsof-command-examples-in-linux/).

**26. last command**

With last command we can watch user’s activity in the system. This command can execute normal user also. It will display complete user’s info like **terminal**, **time**, **date**, **system reboot** or **boot** and **kernel version**. Useful command to troubleshoot.

**# last**

tecmint pts/1 192.168.50.1 Tue Sep 18 08:50 still logged in

tecmint pts/0 192.168.50.1 Tue Sep 18 07:59 still logged in

reboot system boot 2.6.32-279.el6.i Tue Sep 18 07:54 - 11:38 (03:43)

root pts/1 192.168.50.1 Sun Sep 16 10:40 - down (03:53)

root pts/0 :0.0 Sun Sep 16 10:36 - 13:09 (02:32)

root tty1 :0 Sun Sep 16 10:07 - down (04:26)

reboot system boot 2.6.32-279.el6.i Sun Sep 16 09:57 - 14:33 (04:35)

narad pts/2 192.168.50.1 Thu Sep 13 08:07 - down (01:15)

You can use **last** with **username** to know for specific user’s activity as shown below.

**# last tecmint**

tecmint pts/1 192.168.50.1 Tue Sep 18 08:50 still logged in

tecmint pts/0 192.168.50.1 Tue Sep 18 07:59 still logged in

tecmint pts/1 192.168.50.1 Thu Sep 13 08:07 - down (01:15)

tecmint pts/4 192.168.50.1 Wed Sep 12 10:12 - 12:29 (02:17)

**27. ps command**

**ps** command displays about processes running in the system. Below example show **init** process only.

**# ps -ef | grep init**

root 1 0 0 07:53 ? 00:00:04 /sbin/init

root 7508 6825 0 11:48 pts/1 00:00:00 grep init

**28. kill command**

Use **kill** command to terminate process. First find process **id** with **ps** command as shown below and kill process with **kill -9** command.

**# ps -ef | grep init**

root 1 0 0 07:53 ? 00:00:04 /sbin/init

root 7508 6825 0 11:48 pts/1 00:00:00 grep init

**# kill- 9 7508**

**29. rm command**

**rm** command used to remove or delete a file without prompting for confirmation.

**# rm filename**

Using **-i** option to get confirmation before removing it. Using options ‘**-r**‘ and ‘**-f**‘ will remove the file forcefully without confirmation.

**# rm -i test.txt**

rm: remove regular file `test.txt'?

**30. mkdir command example.**

**mkdir** command is used to create directories under Linux.

**# mkdir director yname**

**Beginner Server Administrator Commands**  
**Compiled and Updated by Mark Rais exclusively for** [**ReallyLinux.com**](http://www.reallylinux.com)

|  |  |
| --- | --- |
| **Command** | **Summary Use** |

|  |  |  |
| --- | --- | --- |
| **Command** | **Summary Use** | |
| **arp** | Command mostly used for checking existing Ethernet connectivity and IP address  Most common use: arp  This command should be used in conjunction with the ifconfig and route commands. It is mostly useful for me to check a network card and get the IP address quick. Obviously there are many more parameters, but I am trying to share the basics of server administration, not the whole book of commands. | |
| **df** | Display filesystem information  Most common use: df -h  Great way to keep tabs on how much hard disk space you have on each mounted file system. You should also review our other commands like [file permissions here.](http://www.reallylinux.com/docs/files.shtml) | |
| **du** | Display usage  Most common use, under a specific directory: du -a  Easily and quickly identify the size of files/programs in certain directories. A word of caution is that you should not run this command from the / directory. It will actually display size for every file on the entire Linux harddisk.  This command is also particularly handy if you are checking system resources. Although I provide a number of [Linux networking related commands](http://www.reallylinux.com/docs/networkingadmin.shtml) if you're interested. | |
|  |  | |
| **find** | Find locations of files/directories quickly across entire filesystem  Most common use: find / -name appname -type d -xdev  (replace the word appname with the name of a file or application like gimp)  This is a very powerful command and is best used when running as root or superuser. The danger is that you will potentially look across every single file on every filesystem, so the syntax is very important. The example shown allows you to **search against all directories below / for the appname found in directories but only on the existing filesystem**. It may sound complex but the example shown allows you to find a program you may need within seconds!  Other uses and more complex but beneficial functions include using the -exec or execute a command.  You may also try the commands: locate or try slocate | |
| **ifconfig** | Command line tool to configure or check all network cards/interfaces  Most common uses: ifconfig and also ifconfig eth0 10.1.1.1  Using the plain ifconfig command will show you the details of all the already configured network cards or interfaces. This is a great way to get a check that your network hardware is working properly. You may also benefit from this review of [server configuration](http://www.reallylinux.com/docs/conflin.shtml). Using the many other options of ifconfig such as the one listed allows you to assign a particular interface a static IP address. I only show an example and not a real world command above. Also review some commands for [file permissions here.](http://www.reallylinux.com/docs/files.shtml). Your best bet, if you want to configure your network card using this command is to first read the manual pages. You access them by typing: man ifconfig | |
| **init** | Allows you to change the server bootup on a specific runlevel  Most common use: init 5  This is a useful command, when for instance a servers fails to identify video type, and ends up dropping to the non-graphical boot-up mode (also called runlevel 3).  The server runlevels rely on scripts to basically start up a server with specific processes and tools upon bootup. **Runlevel 5** is the default graphical runlevel for Linux servers. But sometimes you get stuck in a different mode and need to force a level. For those rare cases, the init command is a simple way to force the mode without having to edit the *inittab* file.   Of course, this command does not fix the underlying problem, it just provides a fast way to change levels as needed. For a more permanent correction to the runlevel, edit your /etc/inittab file to state: id:5:initdefault: | |
| **nano** | Easy to use command line editor are always included with most Linux versions and flavors.  One I tend to use for fast easy editing is nano.  Most common uses: nano filename  A real world example for you to get a better sense on how this works: nano /etc/dhcp3/dhcpd.conf This allows you to edit using nano the dhcpd.conf configuration file from the command line.  Maybe you are not up to speed on vi, or never learned how to use emacs?  On most Linux flavors the text editor named **joe** or one named **nano** are available. These basic but easy to use editors are useful for those who need a text editor on the command line but don't know vi or emacs.  Although, I do highly recommend that you learn and use [Vi and Emacs](http://www.reallylinux.com/docs/editors/editor.shtml) editors as well. Regardless, you will need to use a command line editor from time to time. You can also use **cat** and **more** commands to list contents of files, but this is basic stuff found under the basic linux commands listing.  Try: more filename to list contents of the filename. | |
| **netstat** | Summary of network connections and status of sockets  Most common uses: netstat and also netstat |head and also netstat -r  Netstat command simply displays all sockets and server connections. The top few lines are usually most helpful regarding webserver administration. Therefore if you are doing basic webserver work, you can quickly read the top lines of the netstat output by including the |head (pipe and head commands).  Using the -r option gives you a very good look at the network routing addresses. This is directly linked to the route command. | |
| **nslookup** | Checks the domain name and IP information of a server  Most common use: nslookup www.hostname.com  You are bound to need this command for one reason or another. When performing server installation and configuration this command gives you the existing root server IP and DNS information and can also provide details from other remote servers.  Therefore, it is also a very useful security command where you can lookup DNS information regarding a particular host IP that you may see showing up on your server access logs. Note there are some other commands like [file permissions](http://www.reallylinux.com/docs/files.shtml) that may also help. There is a lot more to this command and using the man pages will get you the details by typing: man nslookup | |
| **ping** | Sends test packets to a specified server to check if it is responding properly  Most common use: ping 10.0.0.0 (replace the 10.0.0.0 with a true IP address)  This is an extremely useful command that is necessary to test network connectivity and response of servers. It creates a series of test packets of data that are then bounced to the server and back giving an indication whether the server is operating properly.  It is the first line of testing if a network failure occurs. If ping works but for instance FTP does not, then chances are that the server is configured correctly, but the FTP daemon or service is not. However, if even ping does not work there is a more significant server connectivity issue& like maybe the wires are not connected or the server is turned off!  The outcome of this command is pretty much one of two things. Either it works, or you get the message *destination host unreachable*. It is a very fast way to check even remote servers. | |
| **ps** | Lists all existing processes on the server  Most common uses: ps and also ps -A |more  The simple command will list every process associated with the specific user running on the server. This is helpful in case you run into problems and need to for instance kill a particular process that is stuck in memory. On the other hand, as a system administrator, I tend to use the -A with the |more option.  This will list every process running on the server one screen at a time. Read more of our commands on our [reallylinux.com help page.](http://www.reallylinux.com/docs/consult.shtml) I use ps to quickly check what others are goofing with on my servers and often find that I'm the one doing the dangerous goofing! | |
| **rm** | Removes/deletes directories and files  Most common use: rm -r name (replace name with your file or directory name)  The -r option forces the command to also apply to each subdirectory within the directory. This will work for even non-empty directories. For instance if you are trying to delete the entire contents of the directory x which includes directories y and z this command will do it in one quick process. That is much more useful than trying to use the rmdir command after deleting files! Instead use the rm -r command and you will save time and effort.  You may already have known this but since server administrators end up spending a lot of time making and deleting I included this tip! | |
| **route** | Lists the routing tables for your server  Most common use: route -v  This is pretty much the exact same output as the command netstat -r. You can suit yourself which you prefer to run. I tend to type netstat commands a lot more than just route and so it applies less to my situation, but who knows, maybe you are going to love and use route the most! | |
| **shred** | Deletes a file securely by overwriting its contents  Most common use: shred -v filename (replace filename with your specific file)  The -v option is useful since it provides extra view of what exactly the shred tool is doing while you wait. On especially BIG files this could take a bit of time. The result is that your file is so thoroughly deleted it is very unlikely to ever be retrieved again.  This is especially useful when trying to zap important server related files that may include confidential information like user names or hidden processes. It is also useful for deleting those hundreds of love notes you get from some of the users on your server, another bonus of being a server administrator. :) | |
| **sudo** | The super-user do command that allows you to run specific commands that require root access.  Most common use: sudo command (replace *command* with your specific one)  This command is useful when you are logged into a server and attempt a command that requires super-user or root privileges. In most cases, you can simply run the command through sudo, without having to log in as root. In fact, this is a very beneficial way to administer your server without daily use of the root login, which is potentially dangerous.  Note there are other commands for [file permissions here.](http://www.reallylinux.com/docs/files.shtml) Below is a simple example of the sudo capabilities: sudo cd /root This command allows you to change directories to the /root without having to login as root. Note that you must enter the root password once, when running a sudo command. | |
| **top** | Displays many system statistics and details regarding active processes  Most common use: top  This is a very useful system administrator tool that basically gives you a summary view of the system including number of users, memory usage, CPU usage, and active processes.  Often during the course of a day when running multiple servers, one of my Xwindows workstations just displays the top command from each of the servers as a very quick check of their status and stability. | |
| **touch** | Allows you to change the timestamp on a file.  Most common use: touch filename  Using the basic touch command, as above, will simply force the current date and time upon the specified file. This is helpful, but not often used.  However, another option that I've used in the past when administering servers, is to force a specific timestamp on a set of files in a directory. Read more of our commands on our [reallylinux.com help page.](http://www.reallylinux.com/docs/consult.shtml)   For instance, to force a specific date and time upon all files in a directory, type: touch \*  You can also force a specific date/time stamp using the -t option like this: touch -t200103041200.00 \* The command above will change all files in the current directory to take on the new date of March 4th, 2001 at noon.  The syntax follows this pattern: YYYYMMDDhhmm.ss  YYYY represents the four digit year, then the two digit month, day, hour and minutes. You can even specify seconds as noted above. In any case, this is a useful way to control timestamps on any files on your server. | |
| **traceroute** | Traces the existing network routing for a remote or local server  Most common use: traceroute hostname  (replace hostname with the name of your server such as reallylinux.com)  This is a very powerful network command that basically gives the exact route between your machine and a server. In some cases you can actually watch the network hops from country to country across an ocean, through data centers, etc. Read more of our commands on our [reallylinux.com help page.](http://www.reallylinux.com/docs/consult.shtml)  This comes in handy when trying to fix a network problem, such as when someone on the network can not get access to your server while others can. This can help identify the break or error along the network line. One strong note to you is not to misuse this command!  When you run the traceroute everyone of those systems you see listed also sees YOU doing the traceroute and therefore as a matter of etiquette and respect this command should be used when necessary not for entertainment purposes. A key characteristic of gainfully employed server administrators: knowing when to use commands and when not to use them! | |
| **w** | An extension of the who command that displays details of all users currently on the server  Most common uses: w  This is a very important system admin tool I use commonly to track who is on the server and what processes they are running. It is obviously most useful when run as a super-user.  The default setting for the w command is to show the long list of process details. You can also run the command w -s to review a shorter process listing, which is helpful when you have a lot of users on the server doing a lot of things! Remember that this is different than the ‘who’ command that can only display users not their processes. | |
|  | |  |

## Newbie Commands

**cd –** Changes the current working directory in the command line console.

**exit –** Exits out of the current program, terminates the current command line terminal, or logs you out of a Unix network depending on the context.

**kill –** Terminates the specified running process. The Linux version of Windows’ “End Process” in the task manager.

**ls –** List all of the contents of a specified directory. If no directory is specified, it will use the current directory.

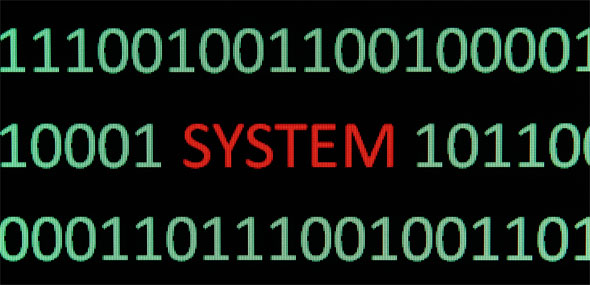
**man –** There’s a running gag in the Linux community that man is the only command you need to know. It stands for manual, and it will give you detailed information on commands and aspects of Linux.

**pwd –** Displays the current working directory for the command line terminal. Good for when you’ve lost track of where you are in your system.

**reboot –** Immediately stops all running processes, shuts down the system, then reboots.

**shutdown –** Stops all running processes and shuts down the system. Parameters can be specified to issue a delayed shutdown or a shutdown at a particular time.

**sudo –** Runs commands as root, which means no limitations due to permissions.



## System Information

**date –** Prints out the current system date and time. Specified parameters can change the format of the output.

**df –** Reports the disk space usage for the file system.

**hostname –** Displays the name of the current host system.

**ps –** Displays information about all of the processes currently running on the system.

**quota –** Displays disk limits and current disk usage for a specified user. Useful when there are multiple users assigned to a particular system.

**top –** Displays all of the top processes in the system, by default sorted by CPU usage.

**uptime –** Reports how long the system has been running since last boot. Extremely useful for servers.



## File Manipulation

**bzip2 –** Compresses specified contents into a .bz2 archive or extracts from a .bz2 archive depending on parameters.

**chmod / chown –** Changes the access permissions of one or more files (chmod) or changes the ownership of a particular file to a new user (chown). Only users with permission or ownership of a file can change that file’s permissions or ownership.

**cp –** Copies files to a new location with a new name depending on the parameters. Can copy directories too, whether recursively (includes all subdirectories) or not.

**find / locate** **–** Searches the system starting at a specific directory and matching all files within that location to a set of conditions laid out by the command parameters. Very useful for quickly finding certain files.

**grep –** Searches through all of the files in a specified location trying to find files that contain lines that match a given string. Returns a list of all the files that scored a match.

**install –** Used in conjunction with Makefiles to copy files from one location to the system. Not to be confused with installing packages from a software repository.

**mkdir / rmdir –** Creates a directory (mkdir) or deletes a specified directory (rmdir). Directories can only be created and deleted within directories that you have permission in.

**mv –** Moves files and directories to another location. Can be used to rename files and directories by keep their source and destination locations the same.

**open –** Opens a specified file using the default system application for files of its type.

**rm –** Remove and remove directory. Used to delete files and directories from the system, whether one at a time or in batch.

**tar –** Creates a .tar archive or extracts from a .tar archive depending on specified parameters.

**zip / unzip –** Creates a .zip archive or extracts from a .zip archive depending on specified parameters.



## Other Noteworthy Commands

**apt-get –** Advanced Packaging Tool. Use this command to install, remove, and configure software packages on your system. For a menu-based version, use aptitude command. Available on Debian-based Linux distributions.

**ftp / sftp –** Connects to a remote FTP server in order to download multiple files.

**wget –** Downloads files from the Internet at the specified URL to your system.

**yum –** Yellowdog Updater, Modified. An open source package manager used to easily install software packages from repositories. Available on RPM-compatible Linux distributions.

**emacs –** One of the most well-known text editors on Unix-like systems.

**nano –** A newbie-friendly command-line text editor that uses keyboard shortcuts to simulate menus.

**vim –** Vim is the successor to Vi, both of which are command line text editors for Unix-like systems. Though Vim is popular, it doesn’t use menus or icons for its interface so it has a reputation for being newbie-friendly.