

Command	Name	Usage	Description
echo	echo	echo Hello	It will print (echo) Hello
ls	list	ls	It lists files in the current directory
		ls -ltr	It lists files in the long format
		ls -F	It lists files and adds markers to files and directories to indicate what they are:
			a trailing / indicates this is a directory an @ indicates a link an * indicates an executable
cd	change directory	cd	It changes directory to your home directory
		cd test	It changes directory to test/
		cd ..	It moves back (up) one directory from or to the parent directory of your current directory
mkdir	make directory	mkdir test	It creates a directory called test
rmdir	remove directory	rmdir test	It removes directory test
cp	copy	cp file1 test/ cp file1 file1.bak	It copies file1 to test directory It makes a backup of file1
rm	remove	rm file1	It removes/deletes file1
mv	move	mv file1 file2 mv file1 test/ cat file1	It renames file1 to file2 It moves file1 into a directory called test It displays all the contents of file1
cat	concatenate	cat file1 file2 > file3	It combines (concatenates) the content of file1 and file2 and redirects (saves) it in file3
less	less	less file1	It views file1 one page at a time
more	more	more file1	It views file1 one page at a time
touch	touch	touch file1	It creates an empty file called file1
nano	nano	nano file1	It creates a new file called file1 and opens it in nano text editor
diff	difference	diff file1 file2	It displays line by line difference between file1 and file2
man	manual	man ls	It displays online manual or help about the ls command
help	help	ls --help	It displays online manual or help about the ls command
pwd	print working directory	pwd	It displays the present working directory, e.g. /user/Shell_scripts_2020

Command	Name	Usage	Description
wc	word count	wc file1	It counts the number of lines (l), words (w) and (characters) in file1
		wc -l file1	It counts the number of lines (l) in file1
		wc -w file1	It counts the number of words (w) in file1
		wc -c file1	It counts the number of (characters) in file1
gzip	G zip	gzip file1	It compresses file1 to produce file1.gz
gunzip	G unzip	gunzip file1.gz	It uncompresses file1.gz to produce file1
history	history	history	It lists all the commands which have been recently used
		history > file1	It saves all the commands which have been recently used in file1
sort	sort	sort file1	This will sort the contents of file1 and display sorted output on the screen
head	head	head file1	It displays the first 10 lines of file1
tail	tail	tail file1	It displays the last 10 lines of file1
grep	grep	grep Hello file1	It searches for the lines containing Hello in file1
		grep -c Hello file1	It gives count or number of lines that contains Hello in file1
find	find	find . -name '*.t' -print	It searches in the current directory (.) and in all its subdirectories for files ending with .t, and writes their names in the output
cut	cut	cut -d, -f 2 file1	It will 'cut out' (extract) the second field (f) (column) from file1 containing commas (,) as the delimiter (d)
uniq	unique	uniq file1	It will filter out <b>adjacent</b> matching lines in file1, i.e. it will remove duplicate lines (NB! File should be sorted before using uniq)

Command	Name	Usage	Description
chmod	change access permission [---][---][---]		
<i>Numerical permissions</i>			
			It provides full access to file1 by [owner/user], [group], and [everyone else]
	7, rwx: read, write, and execute	chmod 777 file1	
	6, rw-: read and write	chmod 766 file1	It provides read and write permission to file1 by [group], and [everyone else]
	5, r-x: read and execute	chmod 775 file1	It provides read and execute permission to file1 by [everyone else]
	4, r--: read-only	chmod 744 file1	It provides read-only permission to file1 by [group], and [everyone else]
	3, -wx: write and execute	chmod 763 file1	It provides write and execute permission to file1 by [everyone else]
	2, -w-: write only	chmod 722 file1	It provides write-only permission to file1 by [group] and [everyone else]
	1, --x: execute only	chmod 771 file1	It provides execute-only permission to file1 by [everyone else]
	0, ---: none	chmod 700 file1	Only [owner/user] can access file1
<i>Letter permissions</i>			
	u: owner/user	u=rx file1	It provides read and execute permission to file1 by [owner/user]
	g: group	go-rwx file1	It removes read (r), write (w) and execute (x) permissions to file1 by [group] and [others]
	a: all	a+x file1	It provides execute (x) permission to file1 by [all/everybody]
	o: others	o=w file1	It provides write (w) permission to file1 by [others]

Symbol	Name	Usage	Description
>	redirect	echo Hello > file1	It will redirect (save) the output of the echo command to file1 and overwrite contents in file1
>>	append redirect	echo Hello >> file1	It will redirect (save) the output of the echo command to file1 and append Hello to the contents in file1
	pipe	wc -l *.txt   sort -n	It will count the number of lines (l) in all files ending with *.txt in the current directory and send the output to the sort command to be sorted numerically (n)

Loops: apply one or more commands separately to each file in a set of files	
Example	Description
<pre>for filename in file1.txt file2.txt file3.txt do head -n 2 \$filename   tail -n 1 done</pre>	For each file in the list (file1.txt, file2.txt, file3.txt), execute the head -n 2 command (i.e. show the first two lines), and pipe this output to the tail -n 1 command (i.e. show the last line of the previous two lines). Note: output will be printed to the terminal.
<pre>for x in *.txt do cp \$x \$x_backup mv \$x_backup test/ done</pre>	For each file in the list (*.txt, i.e. all files ending with .txt), execute the cp command (i.e. make a copy of the file and add '_backup' to the filename), then move (mv) the backup file into the test directory.

Shell scripts: save and re-use commands	
Example	Description
<p>Create a new script by typing <b>nano script.sh</b></p> <p>Add the following text:</p> <pre># Script to find unique names in csv files where names are in the second data field # This script accepts any number of file names as command line arguments # Loop over all files for file in \$@ do echo "Unique names in \$file:" # Extract names cut -d , -f 2 \$file   sort   uniq done</pre> <p>Use Ctrl+O to save the script and Ctrl+X to exit from nano</p> <p>Type <b>script.sh</b> to run the script</p>	<p>Create a new file</p> <p>This will be a comment in the script (i.e. not a command)</p> <p>Another comment</p> <p>Another comment</p> <p>Start the <b>for</b> loop</p> <p><b>do</b> (part of the for loop)</p> <p><b>echo</b> command executed</p> <p>Comment</p> <p><b>cut, sort</b> and <b>uniq</b> commands executed in sequence using pipes  </p> <p><b>done</b> (part of the for loop)</p>