

Basic Linux Command

Command	Description	Example
ls	List directory contents	<ul style="list-style-type: none"> List all including hidden file in list format: <code>\$ ls -al</code> List all files includes directories in their subdirectories recursively: <code>\$ ls -lR</code> List all files with extension only: <code>\$ ls *.*</code>
cd	Change directories	<ul style="list-style-type: none"> Change directory to <code>/var/lib/mysql</code>: <code>\$ cd /var/lib/mysql</code> Change back to home directory: <code>\$ cd ~</code> Return back to the last directory that you were in: <code>\$ cd -</code> To move back to a directory: <code>\$ cd ..</code>
cat	Output a file to screen. Can combine with pipe	<ul style="list-style-type: none"> Output a text file: <code>\$ cat myfile.txt</code> Output a text file and just view any line related to servername: <code>\$ cat myfile.txt grep servername</code>
mv	Change the name of directory, move	<ul style="list-style-type: none"> Move <code>/root/file1.txt</code> to <code>/home/user1/public_html</code> directory: <code>\$ mv /root/file1.txt /home/user1/public_html</code> Rename <code>file1.txt</code> to <code>file2.txt</code>: <code>\$ mv /root/file1.txt /root/file2.txt</code>
cp	Copy files	<ul style="list-style-type: none"> Copy <code>myfile.txt</code> to <code>yourfile.txt</code>: <code>\$ cp myfile.txt yourfile.txt</code> Copy <code>myfile.txt</code> to <code>yourfile.txt</code> including permission, ownership and timestamp: <code>\$ cp -a myfile.txt yourfile.txt</code>
rm	Remove files or directories	<ul style="list-style-type: none"> Remove <code>myfile.txt</code>: <code>\$ rm myfile.txt</code> Remove all PHP files in this directory: <code>\$ rm *.php</code> Remove <code>test1</code> directory and all files underneath it: <code>\$ rm -rf /root/scripts/test1</code>
mkdir	Create directory	<ul style="list-style-type: none"> Create directory called <code>abc</code> under <code>/root</code>: <code>\$ mkdir /root/abc</code> Create directory and all its path if not exist: <code>\$ mkdir /root/abc/set1/set1001/images</code>
pwd	Print working directory	<ul style="list-style-type: none"> Show current working directory: <code>\$ pwd</code>
ln	Create symbolic link/short cut to a file	<ul style="list-style-type: none"> Create a symbolic link called <code>myfile.txt</code> to <code>/root/file1.txt</code>: <code>\$ ln -s /root/file1.txt myfile.txt</code> Create a symbolic link called <code>mydir</code> to directory <code>/root/mydirectory</code>: <code>\$ ln -s /root/mydirectory mydir</code>
clear	Clear the terminal screen	<ul style="list-style-type: none"> Clear SSH screen: <code>\$ clear</code> Another shortcut is using keyboard button <code>Ctrl + L</code>
locate	File locator based on a database. Need to install <code>mlocate</code> package and use <code>updatedb</code> command to create the database	<ul style="list-style-type: none"> Update file location database: <code>\$ updatedb</code> Locate a file called <code>my.cnf</code>: <code>\$ locate my.cnf</code>
find	Search the specified path for files with pattern	<ul style="list-style-type: none"> Search all PHP files under directory <code>/home/user1</code>: <code>\$ find /home/user1 -name *.php -type f</code> Search a directory called <code>includes</code> under <code>/home/user1</code>: <code>\$ find /home/user1 -name includes -type d</code>
whereis	Report all known instance of a command	<ul style="list-style-type: none"> To find where is <code>cp</code> command located: <code>\$ whereis cp</code>
whoami	Report the current user	<ul style="list-style-type: none"> To show what is the user that you currently login as: <code>\$ whoami</code>
more	Output file contents with one page at a time	<ul style="list-style-type: none"> To show output from start of a long log file <code>/var/log/messages</code>: <code>\$ more /var/log/messages</code>
less	Opposite of more command	<ul style="list-style-type: none"> Show output from start with capabilities to scroll up and down of <code>/var/log/messages</code>: <code>\$ less /var/log/messages</code>
diff	Compare 2 files and report all discrepancies	<ul style="list-style-type: none"> Compare <code>/root/report1.txt</code> and <code>/root/report2.txt</code>: <code>\$ diff /root/report1.txt /root/report2.txt</code>

tar	Extract a TAR formatted compressed file	<ul style="list-style-type: none"> • Untar a file called myfile.tar.gz from tar and gzip: <code>\$ tar -xzf myfile.tar.gz</code> • Compress directory /home/user1/myfile to tar and gzip: <code>\$ tar -czf myfile.tar.gz /home/user1/myfile</code>
wget	Download a file using command line	<ul style="list-style-type: none"> • Download the cPanel installer script: <code>\$ wget http://httpdupdate.cpanel.net/latest</code>
df	Report file system disk usage	<ul style="list-style-type: none"> • Show file system free usage: <code>\$ df -h</code>
shutdown	Shutdown or reboot system	<ul style="list-style-type: none"> • Shuts the system down immediately: <code>\$ shutdown -h now</code> • Shuts the system down immediately and reboot afterwards: <code>\$ shutdown -r now</code>
reboot	Reboot system (required root)	<ul style="list-style-type: none"> • Reboot system: <code>\$ reboot</code>
free	Display amount of free system memory	<ul style="list-style-type: none"> • Check free RAM in megabyte: <code>\$ free -m</code>
file	Print description of the file type	<ul style="list-style-type: none"> • Show description of MySQL backup file, /backup/db1.sql: <code>\$ file /backup/db1.sql</code>
kill	Kills specified PID (process ID)	<ul style="list-style-type: none"> • To determine PID, we use ps command. Lets say we want to kill SNMP process: <code>\$ ps aux grep snmp</code> • Kill PID 10765 for SNMP process: <code>\$ kill -9 10765</code>
killall	Kills all process that are instances of the program	<ul style="list-style-type: none"> • Kill all HTTPD process: <code>\$ killall httpd</code>
uptime	Print the system uptime	<ul style="list-style-type: none"> • Print system uptime: <code>\$ uptime</code>
w	Print current system users with load summary	<ul style="list-style-type: none"> • Print system users: <code>\$ w</code>
who	Show who is logged on	<ul style="list-style-type: none"> • Show all logged in users: <code>\$ who</code>
wall	Print message to each user	<ul style="list-style-type: none"> • Warn all active users to ready for system reboot: <code>\$ wall 'This server will be reboot in 1 minute'</code>
top	Print a display of system processes continually	<ul style="list-style-type: none"> • Show active process with load summary: <code>\$ top</code>
logout	Logs the current user off the system	<ul style="list-style-type: none"> • Log off from SSH session: <code>\$ logout</code>
head	Output the first part of files	<ul style="list-style-type: none"> • Show the first 11 line for httpd.conf: <code>\$ head -11 /etc/httpd/conf/httpd.conf</code>
tail	Output the last part of files	<ul style="list-style-type: none"> • Show the last 50 lines of /var/log/messages: <code>\$ tail -50 /var/log/messages</code> • Force to show the last line to monitor /var/log/messages: <code>\$ tail -f /var/log/messages</code>
history	Output list of command that have been used before	<ul style="list-style-type: none"> • Show command history which having mysql command: <code>\$ history grep mysql</code> • Show command history page per page: <code>\$ history more</code>
wc	Print byte, word and line counts	<ul style="list-style-type: none"> • Count the number of users: <code>\$ who wc -l</code> • Count number of network connection for port 80: <code>\$ netstat -noa grep :80 wc -l</code>
init	Upstart process management daemon	<ul style="list-style-type: none"> • Change to single mode: <code>\$ init 1</code> • Change to multi-user mode (normal): <code>\$ init 3</code> • Shutdown: <code>\$ init 0</code> • Reboot: <code>\$ init 6</code>