UNIX / Linux Command Summary

**access()**  
Used to check the accessibility of files

**int**  
Access(pathname, access\_mode)  
Char\* pathname;  
int access-mode;  
The access modes are.  
04 read  
02 write  
01 execute (search)  
00 checks existence of a file

**& operator**  
execute a command as a background process.

**banner**  
prints the specified string in large letters. Each argument may be upto 10 characters long.

**break**  
is used to break out of a loop. It does not exit from the program.

**Cal**  
Produces a calender of the current month as standard output. The month (1-12) and year (1-9999) must be specified in full numeric format.

**Cal** [[ month] year]

**Calendar**  
Displays contents of the calendar file

**case operator**   
The case operator is used to validate multiple conditions.

**Case $string in**

Pattern 1)  
Command list;;  
Command list;;

Pattern 3)  
Command list;;  
easc

**cat**  
(for concatenate) command is used to display the contents of a file. Used without arguments it takes input from standard input <Dtrl d> is used to terminate input.

**cat [filename(s)]**cat > [filename]  
Data can be appended to a file using >>

**Some of the available options are :**Cat [-options] filename(S)  
-s silent about files that  
cannot be accessed  
-v enables display of non printinging characters (except tabs, new lines, form-

**feeds)**-t when used with –v, it causes tabs to be printed as ^I’s  
-e when used with –v, it causes $ to be printed at the end of each line  
The –t and –e options are ignored if the –v options is not specified.

**cd**  
Used to change directories

**chgrp**  
Changes the group that owns a file.  
Chgrp [grou –id] [filename]

**chmod**  
Allows file permissions to be changed for each user. File permissions can be changed only by the owner (s).  
Chmod [+/-][rwx] [ugo] [filename]

**chown**  
Used to change the owner of a file.  
The command takes a file(s) as source files and the login id of another user as the target.  
Chown [user-id] [filename]

**cmp**  
The cmp command compares two files (text or binary) byte-by-byte and displays the first occurrence where the files differ.  
Cmp [filename1] [filename2] -1 gives a long listing

**comm.**  
The comm command compares two sorted files and displays the instances that are common. The display is separated into 3 columns.  
Comm. filename1 filename2  
first displays what occurs in first files but not in the second  
second displays what occurs in second file but not in first  
third displays what is common in both files

**continue statement**  
The rest of the commands in the loop are ignored. It moves out of the loop and moves on the next cycle.

**cp**  
The cp (copy) command is used to copy a file.  
Cp [filename1] [filename2]

**cpio(copy input/output)**  
Utility program used to take backups.  
Cpio operates in three modes:  
-o output  
-i input  
-p pass

**creat()**the system call creates a new file or prepares to rewrite an existing file. The file pointer is set to the beginning of file.  
#include<sys/tyes.h>  
#include<sys/stat.h>  
int creat(path, mode)

**char \*path;**int mode;

**cut**  
used to cut out parts of a file. It takes filenames as command line arguments or input from standard input. The command can cut columns as well as fields in a file. It however does not delete the selected parts of the file.  
Cut [-ef] [column/fie,d] filename  
Cut-d “:” –f1,2,3 filename  
Where –d indicates a delimiter specified within “:”

**df**  
used to find the number of free blocks available for all the mounted file systems.  
#/etc/df [filesystem]

**diff**  
the diff command compares text files. It gives an index of all the lines that differ in the two files along with the line numbers. It also displays what needs to be changed.  
Diff filename1 filename2

**echo**  
The echo command echoes arguments on the command line.  
echo [arguments]  
  
**env**  
Displays the permanent environment variables associated with a user’s login id

**exit command**  
Used to stop the execution of a shell script.

**expr command**  
Expr (command) command is used for numeric computation.  
The operators + (add), -(subtract), \*(multiplu), /(divide), (remainder) are allowed. Calculation are performed in order of normal numeric precedence.

**find**  
The find command searches through directories for files that match the specified criteria. It can take full pathnames and relative pathnames on the command line.  
To display the output on screen the –print option must be specified

**for operator**  
The for operator may be used in looping constructs where there is repetitive execution of a section of the shell program.  
For var in vall val2 val3 val4;

**Do commnds; done**

**fsck**  
Used to check the file system and repair damaged files. The command takes a device name as an argument  
# /etc/fsck /dev/file-system-to-be-checked.

**grave operator**  
Used to store the standard the output of a command in an enviroment variable. (‘)

**grep**  
The grep (global regular expression and print) command can be used as a filter to search for strings in files. The pattern may be either a fixed character string or a regular expression.  
Grep “string” filename(s)

**HOME**  
User’s home directory

**if operator**  
The if operator allows conditional operator

**If expression; then commands; fi**if … then…else… fi  
$ if; then

**commands**efile; then

**commands**fi

**kill**  
used to stop background processes

**In**  
used to link files. A duplicate of a file is created with another name

**LOGNAME**  
displays user’s login name

**ls**  
Lists the files in the current directory   
  
Some of the available options are:  
-l gives a long listing  
-a displays all file{including hidden files

**lp**   
used to print data on the line printer.  
Lp [options] filename(s)

**mesg**   
The mesg command controls messages received on a terminal.  
-n does not allow messages to be displayed on screen  
-y allows messages to be displayed on screen

**mkdir**   
used to create directories

**more**  
The more command is used to dispay data one screenful at a time.  
More [filename]

**mv**  
Mv (move) moves a file from one directory to another or simply changes filenames. The command takes filename and pathnames as source names and a filename or exiting directory as target names.  
mv [source-file] [target-file]

**news**  
The news command allows a user to read news items published by the system administrator.

**ni**  
Displays the contents of a file with line numbers

**passwd**  
Changes the password

**paste**  
The paste command joins lines from two files and displays the output. It can take a number of filenames as command line arguments.  
paste file1 file2

**PATH**  
The directories that the system searches to find commands  
  
**pg**   
Used to display data one page (screenful) at a time. The command can take a number of filenames as arguments.  
Pg [option] [filename] [filename2]…..

**pipe**  
Operator (1) takes the output of one commands as input of another command.  
  
**ps**  
Gives information about all the active processes.

**PS1**The system prompt

**pwd**  
(print working directory) displays the current directory.

**rm**  
The rm (remove) command is used to delete files from a directory. A number of files may be deleted simultaneously. A file(s) once deleted cannot be retrieved.  
rm [filename 1] [filename 2]…

**sift command**   
Using shift $1becomes the source string and other arguments are shifted. $2 is shifted to $1,$3to $2 and so on.

**Sleep**  
The sleep command is used to suspend the execution of a shell script for the specified time. This is usually in seconds.

**sort**  
Sort is a utility program that can be used to sort text files in numeric or alphabetical order  
Sort [filename]

**split**  
Used to split large file into smaller files  
Split-n filename  
Split can take a second filename on the command line.

**su**  
Used to switch to superuser or any other user.

**sync**  
Used to copy data in buffers to files

**system0**  
Used to run a UNIX command from within a C program

**tail**  
The tail command may be used to view the end of a file.  
Tail [filename]

**tar**   
Used to save and restore files to tapes or other removable media.  
Tar [function[modifier]] [filename(s)]

**tee**  
output that is being redirected to a file can also be viewed on standard output.

**test command**   
It compares strings and numeric values.  
The test command has two forms : test command itself If test ${variable} = value then  
Do commands else do commands

**File**  
The test commands also uses special operators [ ]. These are operators following the of are interpreted by the shell as different from wildcard characters.   
Of [ -f ${variable} ]

**Then**Do commands  
Elif  
[ -d ${variable} ]

**then**do commands

**else**do commands

**fi**many different tests are possible for files. Comparing numbers, character strings, values of environment variables.

**time**  
Used to display the execution time of a program or a command. Time is reported in seconds.  
Time filename values  
**tr**  
The tr command is used to translate characters.  
tr [-option] [string1 [string2]]

**tty**   
Displays the terminal pathname

**umask**  
Used to specify default permissions while creating files.

**uniq**   
The uniq command is used to display the uniq(ue) lines in a sorted file.  
Sort filename uniq

**until**  
The operator executes the commands within a loop as long as the test condition is false.

**wall**  
Used to send a message to all users logged in.  
# /etc/wall message

**wait**   
the command halts the execution of a script until all child processes, executed as background processes, are completed.

**wc**  
The wc command can be used to count the number of lines, words and characters in a fine.  
wc [filename(s)]  
**The available options are:**   
wc –[options] [filename]  
-1  
-w  
-c  
while operator  
the while operator repeatedly performs an operation until the test condition proves false.

**$ while**Ø do

**commands**Ø done

**who**   
displays information about all the users currently logged onto the system. The user name, terminal number and the date and time that each user logged onto the system.  
The syntax of the who command is who [options]

**write**  
The write command allows inter-user communication. A user can send messages by addressing the other user’s terminal or login id.  
write user-name [terminal number]

**mkdir - make directories**

Usage

mkdir [OPTION] DIRECTORY

Options

Create the DIRECTORY(ies), if they do not already exist.

 Mandatory arguments to long options are mandatory for short options too.

 -m, mode=MODE  set permission mode (as in chmod), not rwxrwxrwx - umask

 -p, parents  no error if existing, make parent directories as needed

 -v, verbose  print a message for each created directory

 -help display this help and exit

 -version output version information and exit

**cd - change directories**

Use cd to change directories. Type cd followed by the name of a directory to access that directory.Keep in mind that you are always in a directory and can navigate to directories hierarchically above or below.

**mv- change the name of a directory**

Type mv followed by the current name of a directory and the new name of the directory.

 Ex: mv testdir newnamedir

**pwd - print working directory**

will show you the full path to the directory you are currently in. This is very handy to use, especially when performing some of the other commands on this page

**rmdir - Remove an existing directory**

**rm -r**

Removes directories and files within the directories recursively.

**chown - change file owner and group**

Usage

chown [OPTION] OWNER[:[GROUP]] FILE

chown [OPTION] :GROUP FILE

chown [OPTION] --reference=RFILE FILE

Options

Change the owner and/or group of each FILE to OWNER and/or GROUP. With --reference, change the owner and group of each FILE to those of RFILE.

 -c, changes like verbose but report only when a change is made

 -dereference affect the referent of each symbolic link, rather than the symbolic link itself

 -h, no-dereference affect each symbolic link instead of any referenced file (useful only on systems that can         change the ownership of a symlink)

 -from=CURRENT\_OWNER:CURRENT\_GROUP

  change the owner and/or group of each file only if its current owner and/or group match those specified here.  Either  may  be  omitted,  in which case a match is not required for the omitted attribute.

-no-preserve-root do not treat `/' specially (the default)

-preserve-root fail to operate recursively on `/'

-f, -silent, -quiet  suppress most error messages

-reference=RFILE use RFILE's owner and group rather than the specifying OWNER:GROUP values

-R, -recursive operate on [files](http://www.debianhelp.co.uk/commands.htm) and directories recursively

-v, -verbose output a diagnostic for every file processed

The  following options modify how a hierarchy is traversed when the -R option is also specified. If more than one is specified, only the final one  takes effect.

-H     if a command line argument is a symbolic link to a directory, traverse it

-L     traverse every symbolic link to a directory encountered

-P     do not traverse any symbolic links (default)

**chmod - change file access permissions**

Usage

chmod [-r] permissions filenames

 r  Change the permission on files that are in the subdirectories of the directory that you are currently in.        permission  Specifies the rights that are being granted. Below is the different rights that you can grant in an alpha  numeric format.filenames  File or directory that you are associating the rights with Permissions

u - User who owns the file.

g - Group that owns the file.

o - Other.

a - All.

r - Read the file.

w - Write or edit the file.

x - Execute or run the file as a program.

Numeric Permissions:

CHMOD can also to attributed by using Numeric Permissions:

400 read by owner

040 read by group

004 read by anybody (other)

200 write by owner

020 write by group

002 write by anybody

100 execute by owner

010 execute by group

001 execute by anybody

**ls - Short listing of directory contents**

-a        list hidden files

-d        list the name of the current directory

-F        show directories with a trailing '/'

            executable files with a trailing '\*'

-g        show group ownership of file in long listing

-i        print the inode number of each file

-l        long listing giving details about files  and directories

-R        list all subdirectories encountered

-t        sort by time modified instead of name

**cp - Copy files**

cp  myfile yourfile

Copy the files "myfile" to the file "yourfile" in the current working directory. This command will create the file "yourfile" if it doesn't exist. It will normally overwrite it without warning if it exists.

cp -i myfile yourfile

With the "-i" option, if the file "yourfile" exists, you will be prompted before it is overwritten.

cp -i /data/myfile

Copy the file "/data/myfile" to the current working directory and name it "myfile". Prompt before overwriting the  file.

cp -dpr srcdir destdir

Copy all files from the directory "srcdir" to the directory "destdir" preserving links (-poption), file attributes (-p option), and copy recursively (-r option). With these options, a directory and all it contents can be copied to another dir

**ln - Creates a symbolic link to a file.**

ln -s test symlink

Creates a symbolic link named symlink that points to the file test Typing "ls -i test symlink" will show the two files are different with different inodes. Typing "ls -l test symlink" will show that symlink points to the file test.

**locate** - A fast database driven file locator.

**slocate -u**

This command builds the slocate database. It will take several minutes to complete this command.This command must be used before searching for files, however cron runs this command periodically  on most systems.locate whereis Lists all files whose names contain the string "whereis". directory.

**more -** Allows file contents or piped output to be sent to the screen one page at a time

**less -** Opposite of the more command

**cat** - Sends file contents to standard output. This is a way to list the contents of short files to the screen. It works well with piping.

**whereis** - Report all known instances of a command

**wc -** Print byte, word, and line counts

**bg**

bg jobs Places the current job (or, by using the alternative form, the specified jobs) in the background, suspending its execution so that a new user prompt appears immediately. Use the jobs command to discover the identities of background jobs.

**cal month year** - Prints a calendar for the specified month of the specified year.

**cat files** - Prints the contents of the specified files.

**clear -** Clears the terminal screen.

**cmp file1 file2 -** Compares two files, reporting all discrepancies. Similar to the diff command, though the output format differs.

**diff file1 file2** - Compares two files, reporting all discrepancies. Similar to the cmp command, though the output format differs.

**dmesg -** [Prints](http://www.debianhelp.co.uk/commands.htm) the messages resulting from the most recent system boot.

**fg**

fg jobs - Brings the current job (or the specified jobs) to the foreground.

**file files -** Determines and prints a description of the type of each specified file.

**find path -name pattern -print**

Searches the specified path for files with names matching the specified pattern (usually enclosed in single quotes) and prints their names. The find command has many other arguments and functions; see the online documentation.

**finger users -** Prints descriptions of the specified users.

**free  -** Displays the amount of used and free system memory.

**ftp hostname**

Opens an FTP connection to the specified host, allowing files to be transferred. The FTP program provides subcommands for accomplishing [file transfers](http://www.debianhelp.co.uk/commands.htm); see the online documentation.

**head files -** Prints the first several lines of each specified file.

**ispell files -** Checks the spelling of the contents of the specified files.

**kill process\_ids**

kill - signal process\_ids

kill -l

Kills the specified processes, sends the specified processes the specified signal (given as a number or name), or prints a list of available signals.

**killall program**

killall - signal program

Kills all processes that are instances of the specified program or sends the specified signal to all processes that are instances of the specified program.

**mail -** Launches a simple [mail client](http://www.debianhelp.co.uk/commands.htm" \t "_new)

[http://konac.kontera.com/javascript/lib/imgs/grey_loader.gif](http://www.debianhelp.co.uk/commands.htm" \t "_new)

that permits sending and receiving email messages.

**man title**

man section title - Prints the specified man page.

**ping host** - Sends an echo request via TCP/IP to the specified host. A response confirms that the host is operational.

[**reboot**](http://www.debianhelp.co.uk/commands.htm) - Reboots the system (requires root privileges).

**shutdown minutes**

shutdown -r minutes

Shuts down the system after the specified number of minutes elapses (requires root privileges). The -r option causes the system to be rebooted once it has shut down.

**sleep time -** Causes the command interpreter to pause for the specified number of seconds.

**sort files -** Sorts the specified files. The command has many useful arguments; see the online documentation.

**split file -** Splits a file into several smaller files. The command has many arguments; see the online documentation

**sync -** Completes all pending input/output operations (requires root privileges).

**telnet host** - Opens a login session on the specified host.

**top -** Prints a display of system processes that's continually updated until the user presses the q key.

**traceroute host** - Uses echo requests to determine and print a network path to the host.

**uptime -** Prints the [system uptime](http://www.debianhelp.co.uk/commands.htm).

**w -** Prints the current system users.

**wall -** Prints a message to each user except those who've disabled message reception. Type **Ctrl-D** to end the message.

<http://www.oreillynet.com/linux/cmd/>

http://www.mediacollege.com/linux/command/linux-command.html