Linux commands summary

|  |  |  |  |
| --- | --- | --- | --- |
| **File/Dir commands** | **search commands** | **Editors** | **Filters/Pagers** |
| touch  mkdir  ls  cat  rm/rmdir  mv  cp  du  tree | locate  find  grep | vi/vim  nano  pice  emacs  gedit | cut  join  sort  uniq  sed  awk  more  less  head  tail  tee |
| **Backup/Compress** | **Job schedule** | **Misc** | **email** |
| tar  cpio  dd  dump  compress/uncompress  gzip/gunzip  bzip2/bunzip2  zip/unzip | crontab  at | Echo  printf  xargs  tty  talk  exit  set/env | mail  mutt  pine |
| **Network** | **System monitoring** | **Disk management** | **System Administration** |
| Ipconfig  route  ping  traceroute/tracert  netstat | uname  free  uptime  sar  vmstat/iostat/mpstat  df | Fdisk  mount  mkfs  fsck  tune2fs | runlevel  Service/systemctl  ntsysv/ chkconfig  init  telinit |
| **User administration** | **Permissions** | **Help** | **Process management** |
| useradd  usermod  userdel  groupadd  groupmod  groupdel  w/who  finger  chage/chsh  id | chmod  chown  chgrp  chattr  lsattr  setfacl  getfacl | apropos  man  info  <command> -h option  /usr/share/doc/ pages | ps/top/pstree  nice/renice  nohup  fg/bg  jobs  kill/pkill/killall  pgrep  pidof |

* command [<options1><option2>…….] [<argument1> <argument2> …. ]

Ex: ls –l (Command with only one option)

ls –a –l (Command with two options)

ls –al (Command with two options combined)

ls –al /var (Command with combined options and one argument)

ls –al /var /tmp (Command with combined options and two arguments)

ls List contents of a directory.

Ex: ls, ls –l , ls –al, ls –ld, ls –R

touch Change file timestamps to the current time. Make the file if it doesn't exist.

Ex: touch <filename>

cat Displays the file contents

Ex: cat filename.txt

tac Displays the file contents in reverse order

Ex: tac filename.txt

cp Copy a file from one location to another.

Ex: cp file1 file2 Copy file1 to file2;

Ex: cp –R dir1 dir2 : Copy dir1 to dir2

mv Move or rename a file or directory.

Ex: mv <source> <destination>

echo prints the argument on the screen

Ex: echo “hello world”

printf Prints the argument in given format

Ex: printf “hello world” ( new line character is not added by default)

mkdir Make a directory.

Ex: mkdir <directory name> : Makes a directory

Ex *mkdir –p /www/chache/var/log* will create all the directories recursively

pwd Print or list the present working directory with full path.

cdChange the current directory. With no arguments "cd" changes to the users home directory. (cd <directory path>)

rm Delete files (Remove files). (rm –rf <directory/file>)

rmdir Removes a directory.

The directory must be empty. (rmdir <directory>)

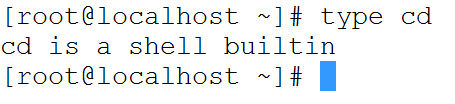
Use rm –r to remove the directory recursively

tree Displays the contents of the given directory in a tree format

* Shell built-ins
* Aliases
* External commands

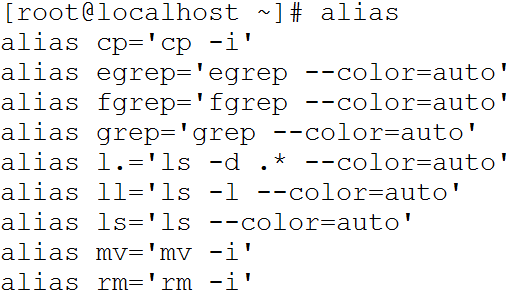
Shell Bult-ins

* Part of your shell
* Different from shell to shell
* cd, pwd, umask etc..
* type <command> will tell you if it is shell built-in



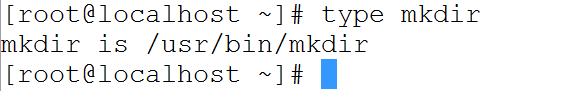
Aliases

* **alias** command displays all the existing aliases



External Commands

* Other commands available from different packages
* ls, mkdir, ifconfig, hostname etc..



cmp Compares the data in the files

diff Finds the differences in the files

join Joins the files contents line by line

paste paste the contents side by side

cut Cuts and display the files of the file by the give delimiter

sort Sort a file alphabetically.

uniq Remove duplicate lines from a sorted file.

grep Searches for the given pattern in the files.

Ex : grep <string> <filename>

grep hello goodbye.txt

find Find files (find <start directory> -name <file name> -print)

Ex: *find /home –name readme -print*

Search for readme starting at home and output full path, “/home" = Search starting at the home directory and proceed through all its subdirectories; "-name readme" = Search for a file named readme "-print" = Output the full path to that file

locate File locating program that uses the slocate database.

Ex: locate –u to create the database,

locate <file/directory> to find file/directory

Compress

gzip zip a file to a gz file.

gunzip unzip a gz file.

zip Compresses a file to a .zip file.

unzip Uncompresses a file with .zip extension.

bzip2 Compress using bzip2

bunzip2 Uncompress using bunzip2

Archiving

tar Archives files and directories. Can store files and directories on tapes.

Ex: tar -zcvf <destination> <files/directories> - Archive copy groups of files. tar –zxvf <compressed file> to uncompress

cpio Can store files on tapes. to/from archives.

Help Commands

apropos To quickly search the available commands for a specific task

man To understand the working of the command and possible options use

info Using the GNU Info System

whereis Locate the binary and man page files for a command. (whereis <program/command>)

which Show full path of commands where given commands reside. (which <command>)

Many tools have a long−style option, `−−help', that outputs usage information about the tool, including the options and arguments the tool takes. Ex: whoami --help

man sections

1 Executable programs or shell commands

2 System calls (functions provided by the kernel)

3 Library calls (functions within program libraries)

4 Special files (usually found in /dev)

5 File formats and conventions eg /etc/passwd

6 Games

7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)

8 System administration commands (usually only for root)

9 Kernel routines [Non standard]

Exit Status

0-255

0 successful

1-255 error

echo $? Displays the exit status of the previous command

System management commands

date Display or change the system date.

Ex: date, date –s “oct 20”

date +%d to display only todays’ date

date +%m to display only the current month

cal displays the current month from the calendar

who See who logged into the system

w See who logged into the system and what are they doing currently with some

advanced options.

printf Print the statements with nice formatting.

exit Exit from the shell.

uptime displays how long the system has been running

cat /proc/cpuinfo displays the process information

free displays RAM available and used on the system

df displays the available partitions and their sizes

fdisk displays or manipulate disk partition table

ifconfig displays network information of the system

ps displays the process running by the current user

ps -ef displays all the process running

top monitors the processes running on systems

route prints routing table information

Performing calculations

$(( )) Perform arithmetic expressions

Ex: echo $((4+5)) prints 9

wc Counts the number of lines in a file

bc Basic calculator

Ex: echo 4+5 | bc prints 9

expr Perform any calculations

let

Network Commands

ping Checks if the remote server is responding

nslookup query Internet name servers interactively

traceroute

tcpdump

iptables

firewall-cmd

ssh Login to remote server using ssh protocol

scp Upload or download files from remote machines

sftp Upload or download files from remote machines. Has many other advantages over scp

telnet login to remote servers using telnet protocol or checks the remote service connectivity

ftp Login to remote server using ftp protocol

File Types

Files -

Directories d

Links l

Block devices b

Character devices c

Pipes p

Sockets s

Regular expressions

. Matches any single character.

? The preceding item is optional and will be matched, at most, once.

\* The preceding item will be matched zero or more times.

+ The preceding item will be matched one or more times.

{N} The preceding item is matched exactly N times.

{N,} The preceding item is matched N or more times.

{N,M} The preceding item is matched at least N times, but not more than M times.

* represents the range if it's not first or last in a list or the ending point of a range in a list.

^ Matches the empty string at the beginning of a line; also represents the

characters not in the range of a list.

$ Matches the empty string at the end of a line.

\b Matches the empty string at the edge of a word.

\B Matches the empty string provided it's not at the edge of a word.

\< Match the empty string at the beginning of word.

\> Match the empty string at the end of word.

VI Editor

* Escape mode
* Insert mode
* Command mode

Escape mode

h Move cursor to left one position

j Move cursor to one line down

k Move cursor to one line up

l Move cursor to right one position

b Move to the beginning of the current word

e Move to the end of the current word

w Move to the beginning of the next word

f<char> Move to the matching character in the current line in forward direction

F<char> Move to the matching character in the current line in reverse direction

gg Takes to the first line of the file

G Takes to the last line of the file

^ Takes to the starting of the current line

$ Takes to the end of the current line

H Takes to the starting of the current screen

M Takes to the middle f the current screen

L Takes to the last of the current screen

dd delete the current line

yy copy the current line

p paste the content below the current line

P(capital) paste the content above the current line

i Enter into insert mode from the current cursor position

I Enter into insert mode and move the cursor to the beginning of the current line

a Enter into insert mode from the next character position

A Enter into insert mode and move the cursor to the end of the current line

dw deletes the current word

dl deletes the current character

yw copies the current word

yl copies the current character

Prefixing a number before the command would execute the command given number of times

2dd deletes 2 lines from current position

2dw deletes 2 words from current position

2yw copies 2 lines from current cursor position

2p pastes the content in buffer 2 times below/after the cursor position

/<pattern> Perform search for the given pattern in forward direction

?<pattern> Perform search for the given pattern in reverse direction

n Finds the next pattern in forward direction

N Finds the next pattern in reverse direction

Command Mode

q Quit or Exit

q! Quit without saving the changes

wq / x Save the changes and Quit

w Save the changes

w <file> Save the change with file name

r <file> Paste the content of the file below the current line

sp <file> Open file in a tab

next Read next file

prev Read previous file

help Show help message

Preferences

set nu #Display line numbers

set ai #auto indentation in programming

set ts=4 # set tab space to 4 rather than default 8

Filters

head displays the last part of the file

tail displays the last part of the file

more Displays the file content page wise

less Display the file content page wise.

Advanced version of more command.

Supports searching and browsing the file.

awk an extremely versatile programming language for working on files

sed stream editor, extremely powerful!

Examples

head /etc/passwd displays first lines by default of /etc/passwd

head -n 20 /etc/passwd displays first 20 lines of /etc/passwd

tail /etc/passwd displays last10 lines by default of /etc/passwd

tail -n 20 /etc/passwd displays last 20 lines of /etc/passwd

tail -f /var/log/messages displays the file in active mode (Ctrl+C to cancel )

more /var/log/messages

* + Enter to scroll down by one line
  + Space to scroll down by page wise
  + q to quit

less /var/log/messages

* + Enter to scroll down by one line
  + Space to scroll down by page wise
  + /<pattern> to search any pattern
  + q to quit

awk -F: '{ print $1 }' /etc/passwd

Displays the first field (usernames) from /etc/passwd

awk -F: '{ print $1 }' /etc/passwd

Displays the third field (user ids) from /etc/passwd

awk -F: '{ if ( $3 > 100 ) print $1 }' /etc/passwd

Displays the user ids only when they are grater than 100 from /etc/passwd