

# SUMMARY OF LINUX SHELL COMMANDS

**Internal commands:** Part of shell (cd, exit, type, help, alias)

**External Commands:** Code resides on disk and executed after fork with exec (clear, ls, passwd, man)

## Basic Shell Commands

Command	Description
<b>echo</b>	Displays text on stdout -n don't append \n -e enables escape sequences -E disable interpretation of backslash escapes (default) -c don't produce any more output
<b>help</b>	Provides detail of internal commands
<b>clear</b>	Clears terminal screen.
<b>exit</b>	To close shell
<b>type</b>	Display information about command type (external/built-in)
<b>logout</b>	To close login shell. <b>Login Shell:</b> when we login, a particular shell starts execution known as login shell
<b>bash</b>	Bash shell
<b>sh</b>	Bourne shell
<b>csh</b>	C-Shell
<b>kch</b>	Korn Shell
<b>tcsh</b>	Tc-Shell
<b>env</b>	Display environment variables
<b>pwd</b>	Shows absolute address of present working directory
<b>passwd</b>	To change user password
<b>man</b>	To view manual pages of different external commands for better understanding. It has 9 sections. -k To search string in all available man pages
<b>who</b>	Shows who is logged in (can be multiple), it also displays "terminal name".
<b>w</b>	Show who is logged on and what they are doing.
<b>whoami</b>	Prints effective username (currently active)
<b>users</b>	Print the user names of users currently logged in to the current host
<b>whatis</b>	Displays command basic purpose (one line description)
<b>whereis</b>	Tells <i>source</i> , <i>binary files</i> and <i>man page</i> file location of external command
<b>which</b>	Gives path of binary file of external and internal command
<b>history</b>	Output the last part of the history list. <i>history [-n]</i>
<b>info</b>	Reads info document of external and internal command
<b>column</b>	Columnize input text -c Specify number of columns -w Specify columns width (1 to 2048)
<b>ls</b>	List directory contents. -a To view hidden files as well -A To view hidden files except '.' and '..' -i Displays inode number -h Displays size in human readable forms in K, M, G instead of bytes -s Shows block count before name (in disk files are saved in blocks) -S Sort all the files and directory w.r.t their sizes and the first file is largest files in all -l List one entry in a line -f List files without sorting as they are stored in directory (it also shows '.' & '..' hidden files) -d List directories themselves not their contents -l Displays files in long listing (7 columns) sorted by names -t Sort by modification time (latest first) -c Sort by status change time (latest first) -u Sort by access time (latest first)
<b>ll</b>	Same as "ls -ls"
<b>touch</b>	Creates 1 or more empty files by touching (updating) modification and access timestamps. If file already exists it updates timestamps: -m For updating modification time only -a For updating access time only -c For updating status change time only
<b>file</b>	display type of file
<b>local</b>	used to declare a variable to be local to a bash function <i>Local [-OPTION] [name[=value]]</i>
<b>read</b>	read a single line from stdin
<b>set</b>	sets shell variables <i>set [-OPTIONS] [arguments]</i>
<b>test</b>	Evaluate condition(s) or make execution of actions dependent on the evaluation of condition(s)

	<b>test[ 'condition' ][ 'condition' ]</b>
<b>expr</b>	Evaluate arguments as an expression: <b>expr `arguments`</b>
<b>cat</b>	To view contents of a simple file on stdout -n To print line numbers as well -s To suppress repeated blank lines -b To number only non-empty lines (overrides -n)
<b>tac</b>	To view contents of file in reverse (last line 1st)
<b>more</b>	To view contents of large files one screen at a time. It also displays % of file displayed and we can't move back up in it. ENTER To move down line-by-line SPACE To move down one screen /str To search "str" in file. Press 'n' to find next Press 'N' to find previous
<b>less</b>	To view contents of large files one page at a time but much better than <b>Navigation</b> Arrow keys, Pgup, Pgdn, ENTER, SPACE (acts as Pgdn), HOME, END /str To search "str" in file. Press 'n' to find next Press 'N' to find previous <b>g, G</b> 'g' moves to start and 'G' moves to end "more".
<b>head</b>	Displays 1st ten lines -n To view 1st n lines
<b>tail</b>	Displays last ten lines -n To view last n lines -f Output appended data as the file grows -c specifies that we want to read n characters not lines
<b>alias</b>	User defined names for commands (arguments are also allowed in alias) alias cls="clear" #makes "cls" an alias for clear command alias #list all aliases
<b>unalias</b>	To remove alias -a Remove all alias definitions
<b>cal</b>	To display calendar -h Won't highlight current-date. -m Display the specified month yyyy Display a calendar for the specified year (e.g. cal 2017) -1 Displays only the current month. -3 Display the previous, current and next month.
<b>date</b>	To display and change (only for root) date [day MON dd mm:hh:ss PKT yyyy]
<b>shutdown</b>	To shut-down or restart shutdown now Shut-down immediately shutdown -r Restarts immediately now shutdown +0 Shut-down immediately shutdown +m Shutdown after m minutes ('+' is optional) shutdown 22:30 Shut-down at 22:30
<b>cp</b>	To copy files/directories -p Preserve permissions while copying (by default permissions can change) -r For directories e.g: cp f1 f2 #f1 is source file and f2 is target-file
<b>rm</b>	To remove files/directories -f ignore non-existent files and arguments, never prompt -r For directories -i For confirmation prompt e.g: rm f1 f2 #will delete both f1 & f2
<b>mv</b>	To move files/directories -i For confirmation prompt e.g: mv f1 f2 #will move f1 to f2 (it is also used to rename file)
<b>mkdir</b>	To make directory file -m set file mode (as in chmod) -p no error if existing, make parent directories as needed
<b>rmdir</b>	To remove directory file -p remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is similar to 'rmdir a/b/c a/b a'
<b>sort</b>	Gets input from stdin and output it on stdout after sorting -b Ignore leading blanks -r for reverse order -t for specifying delimiters (e.g.: -t";") -kn to sort by column n -n for numeric sort -c check for sorted input; do not sort -d Dictionary order -f fold lower case to upper case characters -i consider only printable characters -g compare according to general numerical value (general numeric sort)
<b>length</b>	a string operation to return the number of characters stored in a string

<b>evince</b>	To view PDF and other common document formats
<b>time</b>	Run programs and summarize their system resource usage (shows runtime in seconds). <b>real</b> Total execution time <b>user</b> Time spent in user space <b>sys</b> Time spent in kernel space
<b>lsb_release</b>	Shows basic OS Info. <b>-a</b> Shows all OS details (must be used)
<b>uname</b>	Prints OS name on stdout <b>-a</b> Shows detailed OS info
<b>lscpu</b>	shows detailed CPU specs
<b>readelf</b>	to read ELF files (.o and .out) <b>-a</b> shows all info <b>-s</b> shows symbol table (.symtab) <b>-S</b> shows section header <b>-h</b> shows ELF header <b>-l</b> shows program header
<b>lpr</b>	Line printer prints the contents of specified files to printer
<b>bc</b>	Command line calculator
<b>script</b>	Make typescript of terminal session
<b>wc</b>	print number of lines, word, char counts for each file (Ctrl+D to quit) <b>-l</b> for lines only <b>-w</b> for words only <b>-m</b> for character count only <b>-c</b> byte count
<b>cut</b>	Display selected fields ( <b>-f</b> for fields, <b>-d</b> "delimiter"). Default delimiter is TAB <b>e.g:</b> <code>cut -d":" -f1-3,5 passwd</code> (column 1,2,3,5)
<b>paste</b>	horizontally concatenate files (Separated by TAB)
<b>grep</b>	("General Regular expression Processor") Print lines matching or not matching a pattern. <b>-i</b> for case-insensitive search <b>-v</b> for negation <b>-c</b> print count of lines matching/not matching (for -v)
<b>uniq</b>	Report or omit consecutive repeated/duplicate lines. <b>-c</b> gives line count <b>-u</b> for showing only unique lines <b>-d</b> for showing only duplicated lines
<b>mesg</b>	Permit or deny messages <b>mesg [-y/-n]</b>
<b>split</b>	Split a file into multiple files. Output pieces of FILE to PREFIXaa, PREFIXab, ...; default size is 1000 lines, and default PREFIX is 'x'. With no FILE, or when FILE is -, read standard input. <b>split [OPTION]... [FILE [PREFIX]]</b> <b>-b SIZE</b> put SIZE bytes per output file <b>-C lines</b> put at most SIZE bytes of records per output file
<b>For Comparison and Searching</b>	
<b>comm</b>	Compare 2 sorted files line by line 1st column unique to File1 2nd column unique to File2 3rd column COMMON in both <b>-1</b> suppress column 1 (lines unique to FILE1) <b>-2</b> suppress column 2 (lines unique to FILE2) <b>-3</b> suppress column 3 (lines that appear in both files) <b>--nocheck-order</b> do not check that the input is correctly sorted
<b>cmp</b>	Compare 2 files byte by byte and stops at first difference <b>-l</b> for not stopping on 1st difference (byte values are in octal) <b>Note:</b> All remaining bytes will be different after 1st byte in files
<b>diff</b>	Compare files line by line <b>e.g:</b> <code>diff f1 f2</code> #(I want to make f1 similar to f2) <b>c</b> change <b>a</b> append <b>d</b> delete <b>&lt;</b> is for 1st file <b>&gt;</b> is for 2nd file <b>=</b> is for common lines (in both files) <b>diff -c file1 file2 &gt;new.patch</b> To save differences as a patch file to update 1 <sup>st</sup> file to match 2 <sup>nd</sup> file
<b>locate</b>	To find all location of files by specified name in DB (it don't search in directory hierarchy)
<b>sudo updatedb</b>	to update file DB used by "locate" (updated once per day implicitly)
<b>find</b>	search for files in directory hierarchy <b>-name</b> Finds by name <b>-size</b> Finds by file size (k=Kilobytes, M=Megabytes, G=Gigabytes)

	<b>-atime</b> access time <b>-ctime</b> status change time <b>-mtime</b> modification time <b>type</b> (f = normal files, d = directories, s = sockets, p = named pipes, b=block, c=character, l=soft-link)  <b>EXAMPLES</b> find ~ -mtime 1 Finds files that are modified 1 day ago find ~ -mmin 10 Finds files that are modified 10 min ago find . type f   wc -l Find in the PWD, all the files whose type is regular file and give their count  find / -perm /7000 2>/dev/null for viewing all files with special permissions
--	---

### For Archiving

<b>tar cvf</b>	Create tar file in PWD (1st pass name for archive file then directories and files to archive)
<b>tar tvf</b>	To view .tar files not extract them
<b>tar xvf</b>	To extract .tar files in present working directory
<b>tar xzf</b>	To unzip and extract .tar files in present working directory
<b>gzip</b>	To zip files. Original file is replaced by zip file.(extension = .gz) <b>Note:</b> We can zip tar files to obtain "tar balls"(.tar.gz), commonly used for software distribution
<b>gunzip</b>	To unzip files

### IPC

<b>tee</b>	it reads from stdin and writes to stdout and file(s) <ul style="list-style-type: none"> <li>• It takes all arguments as output file</li> <li>• It doesn't take any input file without input redirection</li> </ul>
<b>mkfifo</b>	it creates named pipes (only)
<b>mknod</b>	<ul style="list-style-type: none"> <li>• it can create named pipes (p).</li> <li>• block special file (b) and</li> <li>• character special file (c)</li> </ul> <pre>mknod -m 0666 file_name type maj min type=b,c,p (block,character,pipe)</pre>

### Vim Editor

Command	Description
<b>sudo apt-get install vim</b>	To install vim editor
<b>vimtutor</b>	For detailed vim lessons
<b>vim +</b>	For opening file in append mode (cursor at last line)
<b>vim +n</b>	Cursor at start of line-n
<b>vim +/string</b>	Cursor on line with 1st occurrence of "string"
<b>ESC</b>	Command mode
<b>ESC+:</b>	Last-line mode
<b>q</b>	To quit vim
<b>i</b>	Start typing before current character
<b>I</b>	Start typing from beginning of current line
<b>a</b>	Start typing after current character
<b>A</b>	Start typing from end of current line
<b>o</b>	Open new line below current line
<b>O</b>	Open new line above current line
<b>h</b>	To move cursor left
<b>L</b>	To move cursor right
<b>K</b>	To move cursor up
<b>j</b>	To move cursor down
<b>gg</b>	To goto 1st line
<b>GG</b>	To goto last line
<b>End, \$</b>	Moves to end of current line
<b>Home, 0</b>	Moves to start of current line
<b>Shift+G</b>	To put prompt at the end of document
<b>u</b>	For undo
<b>Ctrl+r</b>	For redo
<b>ESC+[/,?]</b>	Then write string to search. / For forward search ? For backward search n Find next N For finding in opposite direction
<b>dw</b>	For deleting a word
<b>[n]dd</b>	For deleting a line
<b>[n]yy</b>	For copying line
<b>[n]p</b>	For pasting n times below current line
<b>[n]P</b>	For pasting n times above current line

!	In last-line mode after command to override warning
:wq	in last-line mode to "save & quit"
:w!	to "save" and override warning
:w [filename]	To "save as"
:q!	To quit
:e!	To undo changes since last save
:[n]	To move to nth line
:\$	To move to end of the file
:n1,n2[d,y]	To delete or copy a range of lines. (d= delete, y=copy) :3,6d delete lines 3-6 :3,\$y copy from lines 3-end :9,15y copy lines 9-15
:1,\$ s/search/replace	It will replace only one occurrence in each line of "search" with "replace"
:1,\$ s/search/replace/g	It will replace all occurrences in each line of "search" with "replace"
:set number	To display line numbers
:set nonumber	To remove line numbers
!:command	To execute shell command in last-line mode inside vim editor (will execute only 1 command)
:sh	This will open new bash process to execute commands
<b>For Multi-Filing</b>	
:n	To move to next file
:N	To move to previous file
vim -o filenames	It will open files in multiple horizontal windows
vim -O filenames	It will open files in multiple vertical windows
Ctrl+ww	To move onto next file (if pressed in last file then moves to 1 <sup>st</sup> )

## File Management

Command	Description
lsattr	View extended file attributes
chattr	Change extended file attributes <b>chattr +/-[attr] file</b>
ln	For creating links. -s For soft-links
tty	Display the name of terminal you are using
xterm	to launch ptmx terminal
stty	To change and print terminal line settings. <b>stty attribute value</b> -a To view all attributes <b>stty sane</b> To reset all attributes to their defaults. -isig To off signals on terminal -g To save terminal settings <b>stty -echo</b> Turns off terminal echo <b>stty echo</b> Turns on terminal echo

### File Permissions Management

chown	Change user owner <b>chown :group file</b> (for changing grp owner using chown)
chgrp	Change group owner
chmod	For changing permissions If we write "chmod +r" r will be assigned to u/g/o
umask	To view/change umask -S To view complement of umask in symbolic way
getfacl	Foreach file, getfacl displays the file name, owner, the group, and the Access Control List (ACL). If a directory has a default ACL, getfacl also displays the default ACL.
setfacl	Sets file ACL -m add entry -x remove entry -b clear acl and delete all entries -d add default entry -R add recursive entry

## Process Management

Command	Description
jobs	Lists currently running jobs and their status -l lists process IDs in addition to the normal information -p lists process IDs only

<b>fg [pid]</b> <b>fg %Jid</b>	To move a background process to foreground
<b>bg [pid]</b> <b>bg %Jid</b>	List background jobs or move a process to bg
<b>kill</b>	Send a signal to a job. (default SIGTERM) <b>kill [-s sigspec   -n signal   -sigspec] pid   jobspec</b> <b>kill -l [sigspec]</b> <b>-s sig</b> SIG is a signal name <b>-n sig</b> SIG is a signal number <b>-l</b> list the signal names; if arguments follow <code>`-l'</code> they are assumed to be signal numbers for which names should be listed
<b>ps</b>	Report a snapshot of current process (4 columns) <b>-A or -ax</b> to show all running process <b>-u [username]</b> List processes by user (displays 11 columns) <b>-l</b> displays 14 columns (long listing) <b>-a</b> Select all processes except both session and processes not associated with a terminal.
<b>top</b>	Shows detail process real-time info of top-20 processes, like task manager. Interactive, continuously (refreshes after every 3sec). <b>Press:</b> <b>h</b> for help <b>n</b> to display only [n] processes (0=unlimited) <b>u</b> to display processes of particular user <b>s</b> to change refresh time <b>k</b> to send signal (it 1st asks for PID then signal number/name)
<b>free</b>	Displays amount of free and used memory in the system (6 columns) <b>-k</b> in KB (default) <b>-m</b> in MB <b>-b</b> in bytes <b>-g</b> in GB <b>--tera</b> in tera
<b>vmstat</b>	displays info about virtual memory (6 groups, 17 columns)
<b>uptime</b>	It shows system time, uptime, number of logged in users, load average for last 1,5 and 15 minutes respectively.
<b>watch</b>	Executes a program periodically, showing output in full screen (refreshes every 2sec)
<b>halt</b>	To halt the system.
<b>nice</b>	To run a command with specific NICE(-20 -> 19) value. Only root can use negative nice values. <b>-n</b> add integer N to the niceness (default 10)
<b>renice</b>	Alter priority of running processes. <b>renice [-n] priority [-g -p -u] identifier</b> <b>-n</b> Specify the scheduling priority to be used for the process, process group, or user. When used, it must be the first argument. <b>-g</b> Interpret the succeeding arguments as process group IDs. <b>-p</b> Interpret the succeeding arguments as process IDs (the default). <b>-u</b> Interpret the succeeding arguments as usernames or UIDs. The following command would change the priority of the processes with PIDs 987 and 32, plus all processes owned by the users daemon and root: <b>renice +1 987 -u daemon root -p 32</b>
<b>schedtool</b>	Query and set per-process CPU Scheduling parameters <b>sudo apt-get install schedtool</b> <b>-r</b> lists scheduling policies <ul style="list-style-type: none"> <li>• N: SCHED_NORMAL (prio_min 0, prio_max 0)</li> <li>• F: SCHED_FIFO (prio_min 1, prio_max 99)</li> <li>• R: SCHED_RR (prio_min 1, prio_max 99)</li> <li>• B: SCHED_BATCH (prio_min 0, prio_max 0)</li> <li>• I: SCHED_ISO (policy not implemented)</li> <li>• D: SCHED_IDLEPRIO (prio_min 0, prio_max 0)</li> </ul> <b>-n</b> For changing nice value <b>-p</b> for changing Static Priority <b>-a</b> for changing affinity value <b>-e</b> to execute command with different scheduling parameters <ul style="list-style-type: none"> <li>• schedtool PID</li> <li>• schedtool -[POLICY Letter] PID</li> <li>• schedtool -a [affinity in HEX] PID  e.g: <b>schedtool -a 0x1 3199</b> (0x1,0x2,0x4,0x8,...)</li> <li>• schedtool -n [NICE Val] PID</li> <li>• schedtool -p [Static PRI] PID #for -R -F</li> <li>• For -R -F Static Priority should be mentioned with Policy and in sudo mode</li> </ul>

## User Management

Command	Description
<b>visudo</b>	used to edit /etc/sudoers file
<b>adduser</b>	More interactive and recommended ( <i>sudo adduser user1</i> )
<b>useradd</b>	It is low-level command to add user. And we need to give some extra info as well. Its minimum requirements are: -m to make directory of that user -d To specify directory path (/home/username) <i>useradd -m -d /home/user2 user2</i>
<b>deluser</b>	User we want to delete should be logged out. It don't deletes user HOME Dir
<b>userdel</b>	Low level also deletes HOME directory and files. -r to delete home dir and associated files as well of this user. <i>sudo userdel -r user1</i>
<b>usermod</b>	To modify user info. e.g.: <i>usermod -a -G gp2 user1</i> (makes user1 member of gp2) • If we don't use -a then it will not append new user but overwrite it (that is all previous group members will be removed) -c to change personal info column value <i>sudo usermod -c "Personel Info" user2</i> -s to change default user shell <i>sudo usermod -s /bin/sh user2</i> -l to change username <i>sudo usermod -l user007 user2</i> (new name 1st) -d to change Home Directory -L to lock user (this user can't log in) -U to unlock locked user -g to change primary group -G to change secondary group -a, -- Add the user to the supplementary group(s). Use only with append the -G
<b>groupadd</b>	To add new group. ( <i>sudo groupadd gp1</i> )
<b>groupmod</b>	To modify group. -n is used for changing group name.
<b>groupdel</b>	To delete group. ( <i>sudo groupdel name</i> )
<b>chage</b>	Used to change password expiry info of a user ( <i>sudo chage user2</i> ). -l to view just password setting of particular user
<b>chsh</b>	Used to change default user shell
<b>chfn</b>	Used to change user personal info
<b>finger</b>	shows user info in detail (may have to install it manually)
<b>id</b>	it displays ID (UID) and primary GIDs and groups you belong to gid=primary group, groups = Secondary group
<b>su</b>	(switch user) We can use it to login using any username if we know its password (e.g: su -root) • Using '-' will also give you the target user environment. You will find yourself in the target user HOME Directory and his default login shell

## Disk/Memory Management

Command	Description
<b>Disk Formatting</b>	
<b>man fs</b>	Man page for all commonly used file-systems and their characteristics.
<b>mkfs</b>	to build filesystem (format partitions) • its use is deprecated now there are commands for each FS like <i>mkfs.ext</i> , <i>mkfs.ext2</i> etc. <i>mkfs.&lt;FS_Name&gt;</i> • There should not be any data on partition we want to format if there is take its backup
<b>mount</b>	To mount a partition. <i>mount -t type device /dir</i> <i>mount /dev/sda3 /opt</i> (mounts /dev/sda3 to /opt)
<b>umount</b>	To unmount a partition. ( <i>umount /dev/sda3</i> )
<b>lsblk</b>	Lists info about all available block devices (sda, sr0) • It shows 7-columns by default • To view only selected columns, use -o then enter names of columns (, seperated) <i>lsblk /dev/sda</i> (List info about /dev/sda only.) <i>lsblk -o name,type,fstype,parttype,size,mode /dev/sda</i>
<b>FS Architecture</b>	
<b>e2label</b>	Changes label on ext2/ext3/ext4 file systems. <i>e2label /dev/sda3 "anas3"</i> To assign label <i>e2label /dev/sda3 ""</i> To unassign label
<b>tune2fs</b>	Shows super block info in detail: <i>tune2fs -l /dev/sda1</i>
<b>stat</b>	Lists all i-node block info of a file or device e.g.: inode, permissions, times, size, owners etc.

	<p><b>stat filename/device</b></p> <p><code>stat /etc/passwd</code>  <code>stat -f /dev/sda1</code> (-f shows info about device e.g: /dev/sda1)</p>
<b>df</b>	<p>(Disk free) Displays amount of disk space available on partition/FS</p> <p><b>df -i devices</b></p> <p>If no devices are mentioned then list info for all active partitions.  <b>-i</b> shows info about inodes</p>
<b>du</b>	<p>(Disk usage) Displays how much space a particular file or directory has occupied.</p> <p><b>-h</b> shows size in human readable form K, M, G  <b>du ~</b> Recursively shows sizes of all files, dir, sub-dirs inside ~</p>
<b>lsof</b>	<p>(List of opened files) System Wide File Table</p> <p><b>lsof -p PID</b> (list files opened by PID only)</p>
<b>fuser</b>	<ul style="list-style-type: none"> <li>Identify process using files or sockets.</li> <li>Used to list PIDs and usernames of processes using a specific file</li> </ul> <p><b>-u</b> To show username as well</p> <p><b>fuser -u /etc/passwd</b></p>
<b>Disk Partitioning</b>	
<b>partx</b>	<p>Tell the kernel about the presence and numbering of on-disk partitions</p> <p><b>partx --show /dev/sda</b> (List all partitions on Disk)</p>
<b>dd</b>	<p>Copy a file, converting and formatting according to the operands.</p> <p><b>dd if=/dev/sda bs=512 count=1</b> (Shows contents of zero sector but not human readable)</p> <ul style="list-style-type: none"> <li>It will read file /dev/sda and (if=/dev/sda)</li> <li>reads just 512 bytes and (bs=512)</li> <li>read once and show them (count=1)</li> </ul>
<b>hexdump</b>	<ul style="list-style-type: none"> <li>The hexdump utility is a filter which displays the specified files, or the standard input, if no files are specified, in a user specified format.</li> <li>Makes content of dd readable.</li> </ul> <p><b>-C</b> Display the input offset in hexadecimal, followed by sixteen space-separated, two column, hexadecimal bytes, followed by the same sixteen bytes in %p format enclosed in `` '' characters.</p> <p><b>dd if=/dev/sda bs=512 count=1   hexdump -C</b>  (Shows zero sector in readable HEX format)</p>
<b>fdisk</b>	<p>Manipulate disk partition table (interactive program)</p> <p><b>-l</b> shows info about all the block devices and their partitions</p> <ul style="list-style-type: none"> <li><b>fdisk -l /dev/sda</b> (shows info about sda only)</li> <li><b>fdisk -version</b> (to check fdisk version)</li> </ul> <p>To run fdisk use "<b>fdisk /dev/sda</b>" as root/sudo</p> <p><b>m</b> For help.  <b>p</b> Displays partition table  <b>d</b> To delete partition  <b>n</b> To create new partition  p=primary,e=extended  then write partition digit (1-4)  1st sector (use default)  last sector or size (we should use size in human form by proceeding with +) e.g: +2G  (default for last partition is all remaining disk space)</p> <p><b>q</b> quit without saving changes  <b>w</b> write table to disk and make changes permanent (use it carefully)</p>

## System Programming Commands

Command	Description
<b>make</b>	<p>The make utility will determine automatically which pieces of a large program need to be recompiled, and issue the commands to recompile them.</p> <p><b>make [OPTION]... [TARGET]...</b></p> <ul style="list-style-type: none"> <li><b>-f</b> To specify name of makefile to search for</li> <li><b>-n</b> To tell make to print out what it would have done w/o actually doing it</li> <li><b>-k</b> Tells make to keep going when an error is found, rather than stopping as soon as the first problem is detected.</li> </ul>
<b>ar</b>	<p>The GNU ar program creates, modifies, and extracts from archives. An archive is a single file holding a collection of other files.</p> <p><b>ar -rcs libmymath.a myadd.o mysub.o mydiv.o mymul.o</b></p> <ul style="list-style-type: none"> <li><b>-r</b> Create a new archive <b>ar -r libfirst.a file1.o file2.o</b></li> <li><b>-q</b> Append an object file to an existing archive. <b>ar -q libfirst.a file3.o</b></li> <li><b>-d</b> delete object modules from an existing archive <b>ar -d libfirst.a file2.o</b></li> <li><b>-x</b> extract object modules in your PWD</li> </ul>



	<pre> ar -x /usr/lib/libm.a -t      display table of contents of an archive ar -t /usr/lib/libm.a -c      Without it if an archive is not already existing then a warning will be displayed. -s      To maintain files in particular order w.r.t to functions to avoid errors </pre>
<b>ranlib</b>	ranlib utility generates an index to the contents of an archive and stores it in the archive.
<b>ldd</b>	List dynamic dependencies displays the shared libraries that an executable (or a shared library) requires to run.
<b>ldconfig</b>	Configure dynamic linker run time bindings. Creates necessary links to the most recent shared library versions
<b>objdump</b>	This command can be used to obtain various information, including disassembled binary machine code from an executable file, compiled object or shared library. <pre> -d      To disassemble </pre>
<b>nm</b>	This command lists the set of symbols defined within an object library or executable program
<b>objcopy</b>	Copy and translate object files.
<b>addr2line</b>	Convert addresses into file names and line numbers
<b>gcc</b>	<p>GNU Compiler: <b>gcc [options] file-list</b></p> <pre> -o      Specify the name of executable file (default a.out) -save-temps  To save all intermediate files: (*.i, *.s, *.o, a.out) -E      Perform preprocessing only and generate file with .i extension -S      Generate Assembly code with .s extension for the specific processor -c      Suppress linking phase and keep object files with .o extension -static To force static linking -lxxx   All libraries except std I/O, need to be explicitly linked with -l option. -Lpath  By default, linker looks for libraries in /usr/lib/x86_64/ and /lib/ directories. If you want to link libraries located somewhere else, use -L option -Ipath  By default, preprocessor first searches for include files in directory containing the source file, then in the directory named with -I option to gcc, and finally in /usr/include/ or /usr/include/c++/4.1.1 </pre>
<b>gdb</b>	<p>GNU Compiler we can also specify exe file with it to load it at startup, then we won't have to use <u>file</u> command</p> <pre> -tui   to open gdb in ncurses-interface mode (default 2 panels {code,command}) </pre>
<b>ulimit -c unlimited</b>	To generate core file in case of abnormal termination. <b>gdb -q ex2 core</b>

## GDB Debugger Commands

Command	Description
<b>file</b>	To load program in GDB
<b>attach</b>	to load already running program in gdb using PID
<b>run</b>	to execute loaded program
<b>info registers</b>	to view contents of memory registers
<b>info all-registers</b>	to view contents of all memory registers
<b>info inferiors</b>	to view all current gdb session inferiors (loaded programs) <ul style="list-style-type: none"> <li><b>inferior</b> is used by GDB to manage all loaded programs. Each inferior has a number assigned to it.</li> </ul>
<b>add-inferior</b>	to add new inferior (load another program) <b>add-inferior -exec a.out</b>
<b>info break</b>	to view all breakpoints in focused inferior
<b>list</b>	to view source code inside GDB (it also displays line number)
<b>help</b>	to get help inside gdb <ul style="list-style-type: none"> <li>It shows 12 classes of commands</li> </ul>

```
arif@arif:~/gdb$ gdb -q
(gdb) help
List of classes of commands:

aliases -- Aliases of other commands
breakpoints -- Making program stop at certain points
data -- Examining data
files -- Specifying and examining files
internals -- Maintenance commands
obscure -- Obscure features
running -- Running the program
stack -- Examining the stack
status -- Status inquiries
support -- Support facilities
tracepoints -- Tracing of program execution without stopping the program
user-defined -- User-defined commands
o help all: to view all commands in gdb
o help command: to view info about a gdb command
o help class: to view commands inside a gdb commands class
```

<b>disassemble</b>	used to dump assembly of specified function in AT&T format
<b>backtrace</b>	used to get info about function stack frames (FSF)
<b>finish</b>	It completes execution of current function, returns value to parent function and stop there after copying address of next instruction from FSF to rip.
<b>layout-split</b>	to view an addition assembly code panel in -tui interface mode of gdb

### Breakpoints

<b>break</b>	break command is used to set breakpoint <b>break 10</b> sets breakpoint at line-10 <b>Break main</b> sets breakpoint at 1 <sup>st</sup> line of function main
<b>next/n /ENTER</b>	executes only next instruction (of HLL code) and if that instruction contain function call it will also execute that function code implicitly
<b>continue/c</b>	executes program till end or next breakpoint
<b>ni/si</b>	moves to next instruction of assembly code
<b>disable</b>	to disable a breakpoint temporarily by specifying its number <b>disable 2</b> #disables breakpoint with number-2
<b>print</b>	to view contents of a variable during execution (at breakpoint) <b>print /x i</b> #displays value in HEX <b>print /o i</b> #displays value in Octal <b>print /t i</b> #displays value in Binary <b>print i</b> #displays value in datatype format
<b>whatis</b>	to view datatype of variable
<b>set</b>	to change variable value at breakpoint. It has 2 syntax: o <b>set (i=10)</b> o <b>set variable i=10</b> o <b>set \$rax=9</b> #to change register values precede name with \$

## Commonly used Git Commands

Command	Description
<b>git init</b>	Initializes local git repository in PWD
<b>git clone &lt;Link&gt;</b>	For cloning remote repository for the 1 <sup>st</sup> time
<b>git pull &lt;Link&gt;</b>	Used after clone
<b>git remote rm name</b>	Removes remote repository named "name"
<b>git remote add name &lt;Link&gt;</b>	Adds remote repository named "name"
<b>git push origin master</b>	Pushes master branch to remote repository origin
<b>git branch &lt;name&gt;</b>	Creates new branch named "name"
<b>git branch -l</b>	List all branches
<b>git checkout &lt;branch&gt;</b>	Switches to specified branch
<b>git checkout -b &lt;name&gt;</b>	Creates new branch and switches to it
<b>git branch -m &lt;old&gt; &lt;new&gt;</b>	Rename a branch
<b>git branch -d &lt;branch&gt;</b>	Deletes a branch
<b>git branch -D &lt;branch&gt;</b>	Delete a branch with unmerged changes
<b>git push origin &lt;Branch_Name&gt;</b>	Pushes specified branch to remote repository origin
<b>For COMPARISON</b>	<ul style="list-style-type: none"> <li>• <b>git diff &lt;file&gt;</b> # with staging index</li> <li>• <b>git diff HEAD &lt;file&gt;</b> # with local repo</li> <li>• <b>git diff --statged &lt;file&gt;</b> # cmp file in staging index with local repo</li> <li>• <b>git diff &lt;b1&gt; &lt;b2&gt;</b> # cmp t2o branches</li> </ul>
<b>PATCHING</b>	<ul style="list-style-type: none"> <li>• <b>diff -c file1 file2 &gt;new.patch</b> <ul style="list-style-type: none"> <li>o the differences are identified such that the first file could be modified to make it match the second file</li> </ul> </li> <li>• <b>patch -i new.patch</b> <ul style="list-style-type: none"> <li>o in dir of file1 to change this to file2</li> </ul> </li> </ul>
<b>Config Files</b>	<ul style="list-style-type: none"> <li>• System: /etc/gitconfig</li> <li>• User: ~/.gitconfig</li> </ul>

<b>HEAD</b>	<ul style="list-style-type: none"> <li>• Project: ProjectDir/.git/config</li> <li>• ../git/HEAD (tell branch)</li> <li>• ../git/refs/heads/master (master=branch)</li> <li>• .git/refs/remotes/origin/master (remote master HEAD)</li> </ul>
<b>Excluding files</b>	<ul style="list-style-type: none"> <li>• Project Level: .git/info/exclude</li> <li>• Directory Level: ../gitignore</li> </ul>