**

*AJDEVOPSSOLUTIONS*

***LINUX PRACTICALS-TRAINEDY BY AJAY BONGANI***

**

# System Info Commands

**hostname** - shows the name of the system host.

➜ ~ hostname  
localhost

**hostid** - shows the host id of the system assigned by the OS.

➜ ~ hostid  
0a123456

**date** - shows the current date and time in UTC format.

➜ ~ date  
Wed Jan 19 12:34:56 UTC 2024

**uptime** - shows the elapsed time duration since the machine logged in.

➜ ~ uptime  
12:34:56 up 1 day, 3:45, 2 users, load average: 0.25, 0.20, 0.18

**uname** - unix name.

➜ ~ uname  
Linux

**clear** - clears the screen.

➜ ~ clear

**history** - lists all the commands executed until now.

➜ ~ history  
 1 ls  
 2 cd Documents  
 3 nano file.txt  
 4 gcc program.c -o program  
 5 ./program  
 6 history

**sudo** - Super User Do.

➜ ~ sudo su - USERNAME

**echo $?** - shows the exit status of the last executed command (0 - success, 1-255 - error/failure).

➜ ~ echo $?  
127

**shutdown -r now** - restart the machine immediately (-r restart).

➜ ~ sudo shutdown -r now  
Broadcast message from user@hostname  
 (/dev/pts/0) at 12:34 ...  
  
The system is going down for reboot NOW!

**printenv** - displays all the environment variables of the Linux system.

➜ ~ printenv  
TERM=xterm-256color  
SHELL=/bin/bash  
USER=your\_username  
...

**last** - shows previous logins in the Linux system.

➜ ~ last  
root pts/0 Wed Jan 19 12:34 still logged in  
reboot system boot 5.4.0-96-generic Wed Jan 19 12:33 still running

**systemctl** — System Control: Manage system services using systemd.

➜ ~ systemctl status sshd OR SERVICE SSH STATUS  
● sshd.service - OpenBSD Secure Shell server  
 Loaded: loaded (/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)  
 Active: active (running) since Wed 2024-01-19 12:34:56 UTC; 1 day 3h ago  
 Docs: man:sshd(8)  
 man:sshd\_config(5)  
 Process: 1234 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
 Main PID: 5678 (sshd)  
 Tasks: 1 (limit: 1234)  
 Memory: 2.3M  
 CPU: 12ms  
 CGroup: /system.slice/sshd.service  
 └─5678 /usr/sbin/sshd -D  
  
Jan 19 12:34:56 hostname systemd[1]: Starting OpenBSD Secure Shell server...  
Jan 19 12:34:56 hostname sshd[5678]: Server listening on 0.0.0.0 port 22.  
Jan 19 12:34:56 hostname sshd[5678]: Server listening on :: port 22.  
Jan 19 12:34:56 hostname systemd[1]: Started OpenBSD Secure Shell server.

# File Commands

**touch** - creates an empty file or updates timestamp of the existing file.

* **touch <fileName>** - creates a single empty file.
* **touch <file1> <file2>** - creates file1, file2 empty files.

**cat** - concatenates and displays the contents of files.

* **cat <fileName>** - displays the contents of the file.
* **cat > <fileName>** - creates a new file, allows to input content interactively and redirects inputted content to the created file (> redirection operator).

**head <fileName>** - displays first 10 lines of the file by default.

* **head -n 5 <fileName>** - displays first 5 lines of the file (-n number)

➜ ~ head -n 5 help.txt  
 1. Commands shortcut  
....  
 5. huddle - Connect to Syncup Call

**tail <fileName>** - displays the last 10 lines of the file by default.

* **tail -F <fileName>** - displays contents of the file in real-time even when the file is rotated or replaced (used for log file monitoring).

➜ ~ tail -F mySystem.logs  
echo "I love DevOps"  
echo "Best Linux commands"  
....

**less <fileName>** - used to view large files (log files) in a paginated manner.

**rm** - remove command.

* **rm <fileName>** - removes the file.
* **rm -r <dirName>** - removes files & folders of directory recursively (-r recursive).
* **rm -rf <dirName>** - force remove the files & folders of directory recursively (-f force).
* Example: rm -r ./test

**cp** - copy command.

* **cp <source> <destination>** - copy the files and folders from source to destination.
* **cp -r <dir1> <dir2>** - copy dir1 directory to dir2 directory recursively (-r recursive).
* Example: cp -r ./sourceDir ./destiDir

# File Permission Commands

**ls -l <pathOfFileName>** - shows the permissions of the file.

➜ ~ ls -l .  
total 1016  
-rw-r--r-- 1 ajaydevopssolutions staff 48 Jan 19 21:06 crazy.sh  
-rw-r--r-- 1 ajaydevopssolutions staff 2463 Jan 2 11:25 help

**ls -ld <dirNamePath>** - shows the permissions of the directory.

➜ ~ ls -ld Downloads  
drwx------@ 53 AJDEVOPSSOLUTIONS staff 1696 Jan 19 21:00 Downloads

**chmod <octalNumber> <fileName>** - changes mode/permissions of the file.

* Example: chmod 742 test.txt

**chmod <octalNumber> -R <dirName>** - changes mode/permissions of the directory recursively.

**chown <newUser> <fileName>** - changes the user ownership of a file.

* Example: chown rocky test.txt

**chown <newUser>:<newGroup> <fileName>** - changes the user & group ownerships of a file.

**chgrp <groupName> <fileName/dirName>** - updates the group name for file/directory.

* Example: chgrp mygroup ./test

**getfacl <fileName/dirName>** - shows the file/directory access control list.

➜ ~ getfacl filename.txt  
# file: filename.txt  
# owner: user1  
# group: group1  
user::rw-  
group::r--  
other::r--

**setfacl -m u:<userName>:rwx <fileName/dirName>** - modifies the current acl of the file/directory.

**setfacl -x u:<userName>: <fileName/dirName>** - removes the acl permissions for the file/directory.

**setfacl -m g:<groupName>:rwx <fileName/dirName>** - modifies the group acls for the file/directory.

**setfacl -x g:<groupName>: <fileName/dirName>** - removes the group acl permissions for the file/directory.

# File Permission Octal Numbers

read (r) — 4, write (w)- 2, execute (x) — 1

* Sum the numbers to generate an octal number for setting permissions on a file or directory.

# User Management Commands

[**ac**](https://man7.org/linux/man-pages/man7/man-pages.7.html) **—** Total connect time for all users or specified users.

* The ac command reads the /var/log/wtmp file, which contains binary data about every login, logout, system event, and current status on the system. It gets its data from the wtmp file.
* Display total login time of a specific user.  
  **ac ajay**
* Display total login time for each user.  
  **ac -p**
* Display total login time for each day.  
  **ac -d**
* Display total login time for the current day.  
  **ac -d -p**
* Display login time from a specific log file.  
  **ac -f /var/log/wtmp**

**useradd** - Creates a user account.

* **useradd <userName>** - Creates user account without home & mail spool directories.
* Example: useradd bot
* **useradd -m <userName>** - Creates user account with home & mail spool directories.
* Example: useradd -m bot

**passwd <userName>** - The system generates a password for the user and then stores it in the **/etc/shadow** file.

**userdel** - Deletes User Account.

* **userdel <userName>** - deletes the user from the system.
* **userdel -r <userName>** - deletes the user from the system along with home and mail spool directories.
* Example: userdel -r bot

**/etc/passwd** - Stores information about user accounts.

* **cat /etc/passwd** - displays the complete list of users on that machine.

**/etc/shadow** - stores the password for users in an encrypted format.

* **cat /etc/shadow** - displays the complete list of user passwords on that machine.

**su** - substitute user.

* **su <userName>** - switches to the user mentioned.
* **exit** - to logout from that user.
* Example: su - ram

**usermod** - modify user.

* **usermod -aG <groupName> <userName>** - adds the user to another group (-aG append the user to the group without removing from other groups).
* Example: usermod -aG mygroup ram

**chsh** - change shell.

* **chsh -s /bin/bash <user>** - changes the shell to bash for the user.
* **chsh -s /bin/sh <user>** - changes the shell to sh for the user.
* Example: chsh -s /bin/sh ubuntu

# Group Management Commands

**groupadd <groupName>** - creates the group.

**groupdel <groupName>** - delete the group.

**/etc/group** - stores the information of the groups.

* **cat /etc/group** - displays the complete list of groups on that machine.

**gpasswd <groupName>** - creates a password for the group.

* **gpasswd -a <userName> <groupName>** - adds the user to the group.
* **gpasswd -d <userName> <groupName>** - removes the user from the group.
* **gpasswd -M <userName1>,<userName2>,<userName3> <groupName>** - adds multiple users to the group and removes the existing ones of the group.

# Searching Commands

[**find**](https://www.geeksforgeeks.org/find-command-in-linux-with-examples/) — Search for files/directories based on their names.

* **find . -name <fileName>** - finds the mentioned file if available in the current directory (.(period) represents current directory).
* **find <dirName> -name <fileName>** - finds the mentioned file in the directory.
* **find <dirName> -perm 754** - finds the files in the directory having 754 permission.
* **locate** is faster for finding files by name due to its pre-built database, while **find** is more versatile, allowing complex searches based on various criteria in real-time.

[**locate**](https://ioflood.com/blog/locate-linux-command/)- Search for files/directories based on their names.

* **locate <fileName/dirName>** - locates the file/directory and displays the path.
* Example: locate crazy.txt

# GREP Command — Global Regular Expression Print

* **grep <textToSearch> <fileName>** - used to find text patterns within files.
* **grep -i <textToSearch> <fileName>** - used to find text patterns within the file ignoring the case (-i ignore case).
* **grep -v <textToSearch> <fileName>** - used to find non matching lines of text patterns (-v invert-match).
* **grep -l <textToSearch> <fileNames>** - used to display the matching string file names.
* Example: grep -l welcome crazy.txt
* There are additional commands related to grep.
* [egrep (or grep -E)](https://www.geeksforgeeks.org/fgrep-command-in-linux-with-examples/)
* [fgrep (or grep -F)](https://www.geeksforgeeks.org/fgrep-command-in-linux-with-examples/)
* [zgrep (for compressed files)](https://www.geeksforgeeks.org/zgrep-command-in-linux-with-examples/#:~:text=The%20zgrep%20command%20is%20used,applies%20to%20the%20zgrep%20command.&text=Options%3A,matching%20lines%20for%20each%20file.)
* [zegrep (or zgrep -E for compressed files)](https://www.commandlinux.com/man-page/man1/zegrep.1.html#:~:text=Zgrep%20invokes%20grep%20on%20compressed,necessary%20and%20fed%20to%20grep.)
* [bzgrep (for compressed files)](https://www.geeksforgeeks.org/bzgrep-command-in-linux-with-examples/#:~:text=Syntax%20of%20bzgrep%20command%20in,perform%20a%20case%2Dinsensitive%20search.)
* [ack-grep (Ack)](https://manpages.ubuntu.com/manpages/trusty/man1/ack-grep.1p.html#:~:text=%2D%2Dhelp%2Dtypes%2C%20%2D%2Dhelp,want%20to%20customize%20ack%2D%20grep.)

# Hardware Infomation Commands

**free -h** - Display system memory information in human-readable format (-h).

**df -h** - It displays the disk space usage of mounted file systems.

**du** - Disk usage.

* **du -h** - Display disk usage information in human-readable format.
* **du -sh** - Display the total size of the directory in human-readable format, summarizing the size instead of listing individual file sizes.
* **du -sh <fileName/dirName>** - Displays the total size of the file/directory.

# Connection To Remote System

**ssh** - Secure Shell: Connect to a remote server securely.  
Example: ssh user@remote\_host

**scp** - Securely Copy Files: Copy files between local and remote systems using SSH.  
Example: scp file.txt user@remote\_host:/path

**rsync** - Remote Sync: Synchronize files and directories between systems.  
Example: rsync -avz local\_folder/ user@remote\_host:remote\_folder/

# Network Commands

[**nc**](https://www.computerhope.com/unix/nc.htm) — Simple tcp proxy, network daemon testing

* Example: nc -vz google.com 443

[**ping**](https://www.ibm.com/docs/en/aix/7.2?topic=p-ping-command) **<hostName>** - tests the reachability & responsiveness of the remote host.

* Example: ping google.com -c 2 (-c pings 2 times)

[**dig**](https://www.geeksforgeeks.org/dig-command-in-linux-with-examples/) **<domainName>** - Shows DNS information of the domain.

* Example: dig AJDEVOPSSOLUTIONS.COM

**wget <url>**- Used to retrieve/download files from the internet.

**curl** - client URL.

* **curl <url>** - Used to retrieve/download files from the internet.

**ifconfig** - Display available network interfaces.

**ip addr** - Display and manipulate network interface info.

[**curl ifconfig.me**](https://www.linuxtrainingacademy.com/determine-public-ip-address-command-line-curl/) - Shows the public ip address of the machine.

**netstat -antp**- shows all tcp open ports (-a all, t-tcp, n-active, p protocol).

[**traceroute <url>**](https://www.varonis.com/blog/what-is-traceroute#:~:text=You%20can%20do%20this%20by,%E2%80%9Ctracert%20varonis.com%E2%80%9D.) - traces the route using packets from source to destination host.

# Process Information Commands

**ps** - Process status.

* **ps** - Displays the currently running process.
* **ps -u <userName>**- Displays the process of the username
* **ps -ef** - Displays all the processes of the system.

**top** - Shows the real-time, dynamic view of the running processes of a system.

**kill <pid>** - Gracefully terminates the process pid(-9 forcefull).

**pgrep <processName>** - Shows process id of processes based on name/other criteria.

**bg** - background, sends the process to the background & continues execution without interruption.

**fg** - foreground, brings the process to the foreground and makes it an active process.

**nohup** - no hangup, runs command/script in the background even after the terminal is closed or the user logs out.

* Example: nohup ./script.sh

**<command> &** — Using in last of command runs in **background,** allowing you to continue using the terminal while the command runs asynchronously.

* Example: ./script.sh &

# Archiving File Commands

**tar** - tape archive.

* **tar -cvf <fileName> <directory>** - creates the tar file with the fileName for the directory mentioned (-c create, -v verbose, -f output file name).
* **tar -xvf <sourceTarFileName> -C <destinationDir>** - puts the extracted files into the destination directory (-x extract, -v verbose, -f source tar file name, -C change the folder and download to destination dir).

# Ubuntu Package related Commands

**apt** - Package Manager for Debian-based Linux distributions Eg: Ubuntu.

* **apt** – A newer version of the package manager with colorized output, progress bar and additional functions.
* **apt-get** - Older version and basic package manager.

**apt update** - Updates the package list.

**apt list --installed** - Lists all the installed packages.

* **apt list --installed <packageName>** - shows the package name if it's installed.

**apt show <packageName>** - shows information about a package mentioned.

**apt search <packageName>** - searches and shows the list of packages.

**apt install <packageName>** - installs the required package.

**apt remove <packageName>** - removes the required package.

**apt purge <packageName>** - removes the required package along with its config files.

**Note:** For other package manager just replace “**apt**” with other package manager

# Directory Commands

**pwd** - shows **the present working directory** (abbr. Print Working Directory).

**cd** - change directory.

* **cd ..** - changes to its parent directory (i.e) one level up.
* **cd <dirName>** - change to the directory mentioned.
* **cd ~** or **cd** - changes to the currently logged in user's home directory.
* **cd ../..** - changes the directory two levels up.
* **cd -** - changes to the last working directory.

**mkdir** - make directory.

* **mkdir <dirName>** - creates the directory.
* **mkdir -p <pathOftheDir>** - creates directory with its parent directories if it does not exists (-p parent).

**ls** - lists the files & folders of the directory you are in.

* **ls -a** - lists all files & folders along with hidden files (-a all).
* **ls -al** - lists all files & folders along with hidden files in a formatted manner (-l long listing format).

# Misc Commands

* **man** - Displays the manual page for a specific command. Provides detailed information and usage instructions.
* **sed** - Edits a stream of text by substituting occurrences of a pattern with another.
* **awk** - A powerful programming language for text processing.
* **wc** -(Word Count)
* **ln** -(Create Links):
* **stat <fileName/dirName>** - shows detailed information about the file or directory.
* **cron** - system daemon for managing scheduled tasks.
* **crontab** -Used to create, edit, and manage cron jobs.
* **tree** - Representation of files and directories of a specific directory.
* **echo "sample text" | grep text** - The output of the first command is passed as an input to the second command using the pipe (|) symbol.
* **ls -l | tee file.txt** - Redirects the list to the file.txt and simultaneously displays it in the terminal.
* **echo "sample text" > <fileName>** - Write the content to the file mentioned by overwriting the existing content (> redirection operator).
* **echo "new sample text" >> <fileName>** - Appends the contents to the file mentioned without overwriting the existing content (>> redirection operation).