**Linux Commands**

# **Basic Linux commands**

The following are the basic commands to know while working in the Linux operating system.

**hostname** **:** Displays the hostname of the system

**Hostnamectl :** Get system information including operating system, kernel, and release version

**date** **:** Displays the current system date and time

**ifconfig** **:** Displays the IP and Mac address of the system

**ls**  **:** Lists all files and directories in the present working directory

**ls -R** **:** Lists files in sub-directories as well

**ls -a** **:** Lists hidden files as well

**ls -al** **:** Lists files and directories with detailed information like permissions, size, owner, etc.

**locate file\_name :** Provides the exact location of a file.

**cd** or **cd ~** **:** Navigate to HOME directory

**cd ..** **:** Move one directory level up

**cd dir\_name** **:** Change to a particular directory

**cd /**  **:** Move to the root directory

**cat > filename :** Creates a new file

**cat filename :** Displays all contents of a file

**tail filename** **:** Display the last 10 lines of a file

ls -lt | tail -2 🡪 to get last 2 files in the listed files

**head filename** **:** Display the first 10 lines of a file

**cat file1 file2 > file3** **:** Joins two files (file1, file2) and stores the output in a new file (file3)

**mv filename new\_file\_name** **:** Renames the file to a new file name

**mv filename "new path" :** Moves the file to a new location

**rm filename** **:** Deletes a file

**mkdir directoryname** **:** Creates a new directory in the present working directory

**mkdir “new dir path”** **:** Creates a new directory in the given path

**mv directoryname new\_dir\_name :** Renames the given directory to a new name

**mv directoryname "new path"** **:** Moves the file to a new location

**rmdir** **directoryname** **:** Deletes a given directory

**man commandname** **:** Gives help information of a command

**sort filename :** Sort contents of a file

**history**  **:** Gives a list of all past commands typed in the current terminal session

**clear**  **:** Clears the terminal

**sudo**  **:** Allows regular users to run programs with the security privileges of the superuser or root

**sudo -i** **:** Allows regular users to go to root user

**pr -x :** Divides the file into x columns

**pr -h**  **:** Assigns a header to the file

**pr -n** **:** Denotes the file with Line Numbers

**lpr c** **:** Prints “c” copies of the File

**lp -nc** **:** Prints “c” copies of the File

**lp-d lp-P :** Specifies name of the printer

**apt-get** **:** Installs or updates packages

**mail -s 'subject' 'to-address' -c 'cc-address' -b 'bcc-address'**  **:** sends email with the given subject to email addresses and CC to email addresses or BCC to email addresses

**mail -s "subject" 'to-address' < filename** **:** Sends email with file attachment

# **File Permission commands**

Below commands helps us to work on file permissions

**ls -l** **:** To see file type and access permissions

**r** **:** read permission (has value of 4)

**w** **:** write permission (has value of 2)

**x** **:** execute permission (has value of 1)

**-=** **:** no permission (has value of 0)

**chmod 777 filename** **:** Assign full (read, write, and execute) permissions to everyone

**chmod -R 777 dirname** **:** Assign full permissions to the directory and all sub-directories

**chmod 766 filename** **:** Assign full permissions to the owner but read and write permissions to group and others

**chmod -x filename** **:** Remove the execution permission of any file

**chown username filename** **:** To change the ownership of a file/directory

**chown username:groupname filename** **:** To change the user as well as group of a file or directory

**chown -R username:groupname dirname** **:** Change the owner and group ownership of the directory and all sub-directories

# **Environment Variables commands**

Below commands are useful to set or read environment variables

**echo $VARIABLE\_NAME** **:** Displays the value of a given variable

**env** **:** Displays all environment variables

**VARIABLE\_NAME = variable\_value** **:** Create a new variable with the given value

**unset $VARIABLE\_NAME** **:** Remove a variable

**export VARIABLE\_NAME = variable\_value** **:** To set value of an environment variable

# **User Management commands**

Below commands are useful while working as an administrator

**w** **:** Displays currently logged in users in the system

**sudo adduser username** **:** To add a new user

**sudo passwd -l 'username'** **:** To change the password of a user

**sudo userdel -r 'username'** **:** To remove a newly created user

**sudo usermod -a -G GROUPNAME USERNAME** **:** To add a user to a group

**sudo deluser USER GROUPNAME** **:** To remove a user from a group

**finger** **:** Shows information of all the users logged in

**finger username** **:** Gives information of a particular user

**useradd username** **:** Add a new user account

**userdel -r username** **:** Delete a user account

**usermod [option] username** **:** Change the user account information including, group, home directory, shell, expiration date

**usermod -aG groupname username** **:** Add a user to a specific group

**groupadd groupname** **:** Create a new group

**groupdel groupname** **:** Remove a group

**last** **:** Display information of the last login user

**id** **:** Display UID and GID of the current user

**shutdown -h now** **:** Shut down the system

**reboot** **:** Restart the system

# **Networking commands**

These commands helps us to navigate to the remote Linux machine from local machine

**SSH username@ip-address** or **hostname** **:** To login into a remote Linux machine using SSH

**ping hostname=""** or **=""** **:** To ping and analyze network and host connections

**dir** **:** Display files in the current directory of a remote machine

**cd "directoryname"** **:** To change to a given directory on a remote machine

**get filename** **:** To download a given file from the remote machine to local machine

**put filename** **:** To upload a given file from the local machine to the remote machine

**mget file1 file2 file3** **:** To download multiple files from the remote machine to local machine

**mput file1 file2 file3** **:** To upload multiple files from the local machine to remote machine

**quit** **:** To logout from the remote machine

# **Process commands**

Below commands are useful while working on background processes

**bg** **:** Displays topped or background jobs

**fg** **:** To run a stopped process in the foreground

**top** **:** Displays all active processes running on the machine

**lsof** **:** List all files opened by running processes

**ps** **:** Provides the status of processes running for a user

**pidof** **processname** **:** Gives the Process ID (PID) of a specific process

**ps PID** **:** Gives the status of a particular process using its PID

**ps -ef | grep processname** **:** Display information of specific process

**pstree** **:** Display processes in the tree-like diagram

**kill PID** **:** Kills a process using PID

**killall processname** **:** Kill all processes by name

**nice** **:** Starts a process with a given priority

**renice** **:** Changes the priority of an already running process

**df** **:** Lists free hard disk space on the machine

**free** **:** Lists free RAM on the machine

**free -m** **:** Display free and used memory in the system

**netstat -nap | grep** **PID :**

# **VI Editing Commands**

These commands are useful while working on a file in **vi** mode. Make sure to hit **ESC** button on your keyboard before using these commands.

**i** **:** To insert at cursor (goes into insert mode)

**a** **:** To write after cursor (goes into insert mode)

**A** **:** To write at the end of line (goes into insert mode)

**ESC** **:** To terminate insert mode

**u** **:** To undo the last change

**U** **:** To undo all changes to the entire line

**o** **:** To open a new line (goes into insert mode)

**dd** **:** To delete a single line

**3dd** or **d3d** **:** To delete 3 lines at a time

**D** **:** To delete contents of a line after the cursor

**C** **:** To delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.

**dw** **:** To delete a single word

**4dw** **:** To delete 4 words at a time

**cw** **:** To change a word

**x** **:** To delete a character at the cursor

**r** **:** To replace a character

**R** **:** To overwrite characters from cursor onward

**s** **:** To substitute one character under cursor continue to insert

**S** **:** To substitute entire line and begin to insert at the beginning of the line

**~** **:** To change case of individual character

# **Disk Management Commands**

These commands include, add and remove partitions, mount a partition, check disk space, format partition, etc.

**fdisk -l** **:** List all disk partitions

**fdisk /dev/sda** **:** Create a new partition on /dev/sda device

**mkfs.ext4 /dev/sda1** **:** Format the partition named /dev/sda1

**fsck.ext4 /dev/sda1** **:** Check and repair a filesystem for any error

**mount /dev/sda1 /mnt** **:** Mount any partition to any directory

**df -h** **:** Display free space of mounted file system

**df -i** **:** Display free inodes on the filesystem

**du -hs** **:** Display the size of your current directory

**lsblk** **:** Display information about block devices

**lsusb -tv** **:** Display all USB devices

**hdparm -tT /dev/sda** **:** Perform a read speed test on disk /dev/sda

**badblocks -s /dev/sda** **:** Test for unreadable blocks on disk /dev/sda

# **File Input/Output Redirection Commands**

The **‘<**‘ symbol is used for input(STDIN) redirection and The **‘>**‘ symbol is used for output (STDOUT) redirection.

**program > filename** **:** To redirect output of a command or program to a file by overwriting the existing file contents

**program >> filename** **:** To redirect output of a command or program to a file by appending to the file

**program 0 > ipfilename :** To redirect the standard input of a command or program to a file

**program 1 > outfilename** **:** To redirect the standard output of a command or program to a file

**program 2 > errorfilename** **:** To redirect the standard error of a command or program to a file

**program > filename 2 > &1** **:** To redirect standard error of a command or program to a file of standard output (i.e., both standard error and output are redirected to a single file)

# **Filter Commands:**

Linux has many filter commands such as **grep, awk, sed, cut, wc,** and **spell** to filter the output of any Linux command or contents in a file. To use filter command with another Linux command, concatenate with pipe (|) symbol. Ex. *echo “Hello World” | grep “Hello”*

**cat filename | grep string**

or

**grep string filename :** Searches a specific string and displays matched string data

**grep -options string filename :** Searches a specific string and filters the output as per options specified.

Various options include:

**i :** Performs a case-insensitive (upper or lower case) search for a specific string

**c** **:** Displays count of matching lines

**o** **:** Displays matched string line

**v :** Displays un-matched string line

**l :** Displays file names in the current directory that match a pattern

**n :** Displays line number and the matching line

**grep "string1|string2" filename :** Searches and displays multiple strings in a file

**grep ^string filename :** Searches all contents ending with a given string in a file

**awk '{action}' filename :**   
**awk '{pattern; action}' filename :**

**awk '{print $1}' filename :** Displays the first field ($1) separated by a space in the file.

**echo "string" | awk '{$2="string2"; print $0}' :** Prints output on the given pattern by replacing the second field ($2) with "string2" and displays the whole line ($0)

**sed 's/string/substitute/option' filename :** Replaces a string with the substitute string using the given options in a file

Various options are:

**g :** replaces all occurrences of matching string with a new string

**sed 'linenumberd' filename :** Deletes a specific line in the file

**sed -n 'linenumber s/str/str1/p' filename :** Replaces the string by specifying a line number and only prints the replaced line

**cut -options filename :** Extracts specific lines as per given options

Various options include:

**b :** Cuts section or content using a specified byte or a range of bytes, ex: *cut -b 1 textfile*

**c :** Extracts a string by specifying positions of characters, ex: *cut -c 1,3,5 textfile*

**-c n- :** Extract characters from nth position till end of the line

**-c -n :** Extract characters from starting of the line till nth position

**-c m-n :** Extract characters from mth position to nth position

**f :** Extracts a string by specifying field numbers

**d :** Extracts a string delimited by a character ex: *cut -d “ ” -f 1 textfile*

# **String Manipulation Commands:**

**variableName = ’string’** **:** To assign a string to a variable

**echo $variableName** **:** To print variable value

**${var1}${var2}** or **$var1$var2** or **"$var1""$var2" :** To concatenate multiple strings in variables

**${#variablename}** **:** Returns the length of string in a variable

**${string//oldstring/newstring}** **:** Replaces the substring of a string with a new string. Special characters in the old and new strings should be escaped by symbol “\”

**${string:position}** **:** Returns a substring starting from a given position till end

**${string:position:number}** **:** Returns a number of characters starting from the given position (if position value is < 0 then it returns the entire string, if the number value is < 0 then it throws error)

**${string#substring}** **:** Deletes the shortest substring match from front of a string, to search multiple characters ending with a dot, we can give the substring as ***\*.*** (asterisk and dot)

**${string%substring}** **:** Deletes the shortest substring match from back of a string

**${string##substring}** **:** Deletes the longest substring match from front of a string

**${string%%substring}** **:** Deletes the longest substring match from back of a string

stringZ=abcABC123ABCabc

# |----| shortest

# |----------| longest

echo ${stringZ#a\*C} # 123ABCabc

# Strip out shortest match between 'a' and 'C'.

echo ${stringZ##a\*C} # abc

# Strip out longest match between 'a' and 'C'.

# You can parameterize the substrings.

X='a\*C'

echo ${stringZ#$X} # 123ABCabc

echo ${stringZ##$X} # abc

# As above.

# **Package Management Commands:**

In this section, we will show a list of all commands to install, remove and manage packages in Linux.

**apt-get install packagename** **:** Install the package on Debian based distributions

**apt-get remove packagename** **:** Remove a package on Debian based distributions

**dpkg -l | grep -i installed** **:** Get a list of all packages on Debian based distributions

**dpkg -i packagename.deb** **:** Install .deb package

**apt-get update** **:** Update the repository on Debian based distributions

**apt-get upgrade packagename** **:** Upgrade a specific package on Debian based distributions

**apt-get autoremove** **:** Remove all unwanted packages on Debian based distributions

**yum install packagename** **:** Install the package on RPM-based distributions

**yum remove packagename** **:** Remove a package on RPM-based distributions

**yum update** **:** Update all system packages to the latest version on RPM-based distributions

**yum list --installed** **:** List all installed packages on RPM-based distributions

**yum list --available** **:** List all available packages on RPM-based distributions

# **Compress and Uncompress Commands**

Tar, Zip, and Unzip are the most popular command-line utility in Linux used to compress and uncompress files and directories.

**tar -cvf filename.tar filename** **:** Compress a file in the Tar archive

**tar -xvf filename.tar** **:** Uncompress a Tar file

**tar -tvf filename.tar** **:** List the content of the Tar file

**tar -xvf filename.tar file1.txt** **:** Untar a single file from Tar file

**tar -rvf filename.tar file2.txt** **:** Add a file to the Tar file

**zip filename.zip filename** **:** Compress a single file to a zip

**zip filename.zip file1.txt file2.txt file3.txt** **:** Compress multiple files to a zip

**zip -u filename.zip file4.txt** **:** Add a file to a zip file

**zip -d filename.zip file4.txt** **:** Delete a file from a zip file

**unzip -l filename.zip** **:** Display the content of zip archive file

**unzip filename.zip** **:** Unzip a file

**unzip filename.zip -d /dirname** **:** Unzip a file to a specific directory

# **Combo Commands:**

**ls -l | head -2 :** List top 2 files

**ls -l | tail -2** **:** List bottom 2 files

**ls -t | tail -n +3** **:** Lists all files except 2 latest files

**ls -t | head -n 2 :** Lists all files except 3 old files

**ls -t | tail -n +3 | xargs rm :** Removes all files except 2 latest files

# **Arrays:**

**arrayname=(value1 value2)** **:** Creates an array variable using variables

**arrayname=( $( command) )** **:** Creates an array variable from the output of a Linux command

**${arrayname[index]}** **:** To get a specific item in an array using its index number. Indexing always starts from 0

**${arrayname[\*]}** or **${arrayname[@]}** **:** To get all items in an array

# **If Statement:**

The syntax of an if statement is the following:

if [[ CONDITION\_1 ]] && [[ CONDITION\_2 ]]; then

COMMANDS\_1

elif [ CONDITION\_3]; then

COMMANDS\_2

else

COMMANDS\_3

fi

Nested IF

if CONDITION\_1

then

if CONDITION\_2

then

COMMANDS\_1

else

COMMANDS\_2

fi

else

COMMANDS\_3

fi

----------------------

echo -n "Enter Number: "

read x

if [ $((x%2)) == 0 ]; then

if [ $x == 0 ]; then

echo "Number is Zero"

else

echo "Number is Even"

fi

else

echo "Number is Odd"

fi

-----------------------

if [[ $x -ge $y ]] && [[ $x -ge $z ]]; then

echo "x is greatest"

fi

for FILE in ${@} do if [[ ! -f $FILE ]] then echo "The file ${FILE} does not exist!" fi done

# **Loops:**

**for file in \*.txt ; do wc -l $file ; done**