

1. File and Directory Operations Commands

File and directory operations are fundamental in working with the Linux operating system. Here are some commonly used File and Directory Operations commands:

Command	Description	Options	Examples
ls	List files and directories.	<ul style="list-style-type: none">• -l: Long format listing.• -a: Include hidden files hidden ones• -h: Human-readable file sizes.	<ul style="list-style-type: none">• ls -l displays files and directories with detailed information.• ls -a shows all files and directories, including• ls -lh displays file sizes in a human-readable format.
cd	Change directory.		<ul style="list-style-type: none">• cd /path/to/directory changes the current directory to the specified path.
pwd	Print current working directory.		<ul style="list-style-type: none">• pwd displays the current working directory.
mkdir	Create a new directory.		<ul style="list-style-type: none">• mkdir my_directory creates a new directory named “my_directory”.
rm	Remove files and directories.	<ul style="list-style-type: none">• -r: Remove directories recursively.• -f: Force removal without confirmation.	<ul style="list-style-type: none">• rm file.txt deletes the file named “file.txt”.• rm -r my_directory deletes the directory “my_directory” and its contents.• rm -f file.txt forcefully deletes the file

Command	Description	Options	Examples
cp	Copy files and directories.	<ul style="list-style-type: none"> • -r: Copy directories recursively. 	<p>“file.txt” without confirmation.</p> <ul style="list-style-type: none"> • cp -r directory destination copies the directory “directory” and its contents to the specified destination. • cp file.txt destination copies the file “file.txt” to the specified destination.
mv	Move/rename files and directories.		<ul style="list-style-type: none"> • mv file.txt new_name.txt renames the file “file.txt” to “new_name.txt”. • mv file.txt directory moves the file “file.txt” to the specified directory.
touch	Create an empty file or update file timestamps.		<ul style="list-style-type: none"> • touch file.txt creates an empty file named “file.txt”.
cat	View the contents of a file.		<ul style="list-style-type: none"> • cat file.txt displays the contents of the file “file.txt”.
head	Display the first few lines of a file.	<ul style="list-style-type: none"> • -n: Specify the number of lines to display. 	<ul style="list-style-type: none"> • head file.txt shows the first 10 lines of the file “file.txt”.

Command	Description	Options	Examples
tail	Display the last few lines of a file.	<ul style="list-style-type: none"> • -n: Specify the number of lines to display. 	<ul style="list-style-type: none"> • head -n 5 file.txt displays the first 5 lines of the file “file.txt”. • tail file.txt shows the last 10 lines of the file “file.txt”. • tail -n 5 file.txt displays the last 5 lines of the file “file.txt”.
ln	Create links between files.	<ul style="list-style-type: none"> • -s: Create symbolic (soft) links. 	<ul style="list-style-type: none"> • ln -s source_file link_name creates a symbolic link named “link_name” pointing to “source_file”.
find	Search for files and directories.	<ul style="list-style-type: none"> • -name: Search by filename. • -type: Search by file type. 	<ul style="list-style-type: none"> • find /path/to/search -name “*.txt” searches for all files with the extension “.txt” in the specified directory.

2. File Permission Commands

File permissions on Linux and Unix systems control access to files and directories. There are three basic permissions: read, write, and execute. Each permission can be granted or denied to three different categories of users: the owner of the file, the members of the file’s group, and everyone else.

Here are some file permission commands:

Command	Description	Options	Examples
chmod	Change file permissions.	<ul style="list-style-type: none"> • u: User/owner permissions. • g: Group permissions. 	<ul style="list-style-type: none"> • chmod u+rw file.txt grants read, write, and execute permissions to the owner of the file.

Command	Description	Options	Examples
chown	Change file ownership.	<ul style="list-style-type: none"> • o: Other permissions. • +: Add permissions. • -: Remove permissions. • =: Set permissions explicitly. 	<ul style="list-style-type: none"> • chown user file.txt changes the owner of “file.txt” to the specified user.
chgrp	Change group ownership.		<ul style="list-style-type: none"> • chgrp group file.txt changes the group ownership of “file.txt” to the specified group.
umask	Set default file permissions.		<ul style="list-style-type: none"> • umask 022 sets the default file permissions to read and write for the owner, and read-only for group and others.

3. File Compression and Archiving Commands

Here are some file compression and archiving commands in Linux:

Commands	Description	Options	Examples
tar	Create or extract archive files.	<ul style="list-style-type: none"> • -c: Create a new archive. 	<ul style="list-style-type: none"> • tar -czvf archive.tar.gz files/ creates a compressed tar archive named “archive.tar.gz”

Commands	Description	Options	Examples
		<ul style="list-style-type: none"> • -x: Extract files from an archive. • -f: Specify the archive file name. • -v: Verbose mode. • -z: Compress the archive with gzip. • -j: Compress the archive with bzip2. 	containing the files in the “files/” directory.
gzip	Compress files.	<ul style="list-style-type: none"> • -d: Decompress files. 	<ul style="list-style-type: none"> • gzip file.txt compresses the file “file.txt” and renames it as “file.txt.gz”.
zip	Create compressed zip archives.	<ul style="list-style-type: none"> • -r: Recursively include directories. 	<ul style="list-style-type: none"> • zip archive.zip file1.txt file2.txt creates a zip archive named “archive.zip” containing “file1.txt” and “file2.txt”.

4. Process Management Commands

In Linux, process management commands allow you to monitor and control running processes on the system. Here are some commonly used process management commands:

Commands	Description	Options	Examples
ps	Display running processes.	<ul style="list-style-type: none"> • -aux: Show all processes. 	<ul style="list-style-type: none"> • ps aux shows all running processes with detailed

Commands	Description	Options	Examples
top	Monitor system processes in real-time.		<p>information.</p> <ul style="list-style-type: none"> • top displays a dynamic view of system processes and their resource usage.
kill	Terminate a process.	<ul style="list-style-type: none"> • -9: Forcefully kill a process. 	<ul style="list-style-type: none"> • kill PID terminates the process with the specified process ID.
pkill	Terminate processes based on their name.		<ul style="list-style-type: none"> • pkill process_name terminates all processes with the specified name.
pgrep	List processes based on their name.		<ul style="list-style-type: none"> • pgrep process_name lists all processes with the specified name.
grep	used to search for specific patterns or regular expressions in text files or streams and display matching lines.	<ul style="list-style-type: none"> • -i: Ignore case distinctions while searching. • -v: Invert the match, displaying non-matching lines. 	<ul style="list-style-type: none"> • grep -i "hello" file.txt • grep -v "error" file.txt • grep -r "pattern" directory/ • grep -l "keyword" file.txt

Commands	Description	Options	Examples
		<ul style="list-style-type: none"> • -r or -R: Recursively search directories for matching patterns. • -l: Print only the names of files containing matches. • -n: Display line numbers alongside matching lines. • -w: Match whole words only, rather than partial matches. • -c: Count the number of matching lines instead of displaying them. • -e: Specify multiple patterns to search for. • -A: Display lines after the matching line. • -B: Display lines before the matching line. • -C: Display lines both before and after 	<ul style="list-style-type: none"> • grep -n "pattern" file.txt In these examples we are extracting our desired output from filename (file.txt)

Commands	Description	Options	Examples
		the matching line.	

5. System Information Commands

In Linux, there are several commands available to gather system information. Here are some commonly used system information commands:

Command	Description	Options	Examples
uname	Print system information.	<ul style="list-style-type: none"> • -a: All system information. 	<ul style="list-style-type: none"> • uname -a displays all system information.
whoami	Display current username.		<ul style="list-style-type: none"> • whoami shows the current username.
df	Show disk space usage.	<ul style="list-style-type: none"> • -h: Human-readable sizes. 	<ul style="list-style-type: none"> • df -h displays disk space usage in a human-readable format.
du	Estimate file and directory sizes.	<ul style="list-style-type: none"> • -h: Human-readable sizes. • -s: Display total size only. 	<ul style="list-style-type: none"> • du -sh directory/ provides the total size of the specified directory.
free	Display memory usage information.	<ul style="list-style-type: none"> • -h: Human-readable sizes. 	<ul style="list-style-type: none"> • free -h displays memory usage in a human-readable format.
uptime	Show system uptime.		<ul style="list-style-type: none"> • uptime shows the current system uptime.

Command	Description	Options	Examples
lscpu	Display CPU information.		<ul style="list-style-type: none"> • lscpu provides detailed CPU information.
lspci	List PCI devices.		<ul style="list-style-type: none"> • lspci List PCI devices.
lsusb	List USB devices.		<ul style="list-style-type: none"> • lsusb lists all connected USB devices.

6. Networking Commands

In Linux, there are several networking commands available to manage and troubleshoot network connections. Here are some commonly used networking commands:

Command	Description	Examples
ifconfig	Display network interface information.	<ul style="list-style-type: none"> • ifconfig shows the details of all network interfaces.
ping	Send ICMP echo requests to a host.	<ul style="list-style-type: none"> • ping google.com sends ICMP echo requests to “google.com” to check connectivity.
netstat	Display network connections and statistics.	<ul style="list-style-type: none"> • netstat -tuln shows all listening TCP and UDP connections.
ss	Display network socket information.	<ul style="list-style-type: none"> • ss -tuln shows all listening TCP and UDP connections.

Command	Description	Examples
ssh	Securely connect to a remote server.	<ul style="list-style-type: none"> • ssh user@hostname initiates an SSH connection to the specified hostname.
scp	Securely copy files between hosts.	<ul style="list-style-type: none"> • scp file.txt user@hostname:/path/to/destination securely copies “file.txt” to the specified remote host.
wget	Download files from the web.	<ul style="list-style-type: none"> • wget http://example.com/file.txt downloads “file.txt” from the specified URL.
curl	Transfer data to or from a server.	<ul style="list-style-type: none"> • curl http://example.com retrieves the content of a webpage from the specified URL.

7. IO Redirection Commands

In Linux, IO (Input/Output) redirection commands are used to redirect the standard input, output, and error streams of commands and processes. Here are some commonly used IO redirection commands:

Command	Description
cmd < file	Input of cmd is taken from file.
cmd > file	Standard output (stdout) of cmd is redirected to file.
cmd 2> file	Error output (stderr) of cmd is redirected to file.
cmd 2>&1	stderr is redirected to the same place as stdout.
cmd1 <(cmd2)	Output of cmd2 is used as the input file for cmd1.

Command	Description
<code>cmd > /dev/null</code>	Discards the stdout of cmd by sending it to the null device.
<code>cmd &> file</code>	Every output of cmd is redirected to file.
<code>cmd 1>&2</code>	stdout is redirected to the same place as stderr.
<code>cmd >> file</code>	Appends the stdout of cmd to file.

8. Environment Variable Commands

In Linux, environment variables are used to store configuration settings, system information, and other variables that can be accessed by processes and shell scripts. Here are some commonly used environment variable commands:

Command	Description
export VARIABLE_NAME=value	Sets the value of an environment variable.
echo \$VARIABLE_NAME	Displays the value of a specific environment variable.
env	Lists all environment variables currently set in the system.
unset VARIABLE_NAME	Unsets or removes an environment variable.
export -p	Shows a list of all currently exported environment variables.
env VAR1=value COMMAND	Sets the value of an environment variable for a specific command.
printenv	Displays the values of all environment variables.

9. User Management Commands

In Linux, user management commands allow you to create, modify, and manage user accounts on the system. Here are some commonly used user management commands:

Command	Description
who	Show who is currently logged in.
sudo adduser username	Create a new user account on the system with the specified username.
finger	Display information about all the users currently logged into the system, including their usernames, login time, and terminal.
sudo deluser USER GROUPNAME	Remove the specified user from the specified group.
last	Show the recent login history of users.
finger username	Provide information about the specified user, including their username, real name, terminal, idle time, and login time.
sudo userdel -r username	Delete the specified user account from the system, including their home directory and associated files. The -r option ensures the removal of the user's files.
sudo passwd -l username	Lock the password of the specified user account, preventing the user from logging in.
su - username	Switch to another user account with the user's environment.
sudo usermod -a -G GROUPNAME USERNAME	Add an existing user to the specified group. The user is added to the group without removing them from their current groups.

Shortcuts Commands

VI Shortcuts Commands:

Command	Description
cw	Change the current word. Deletes from the cursor position to the end of the current word and switches to insert mode.
dd	Delete the current line.
x	Delete the character under the cursor.
R	Enter replace mode. Overwrites characters starting from the cursor position until you press the Escape key.
o	Insert a new line below the current line and switch to insert mode.
u	Undo the last change.
s	Substitute the character under the cursor and switch to insert mode.
dw	Delete from the cursor position to the beginning of the next word.
D	Delete from the cursor position to the end of the line.
4dw	Delete the next four words from the cursor position.
A	Switch to insert mode at the end of the current line.
S	Delete the current line and switch to insert mode.
r	Replace the character under the cursor with a new character entered from the keyboard.
i	Switch to insert mode before the cursor.

Command	Description
3dd	Delete the current line and the two lines below it.
ESC	Exit from insert or command-line mode and return to command mode.
U	Restore the current line to its original state before any changes were made.
~	Switch the case of the character under the cursor.
a	Switch to insert mode after the cursor.
C	Delete from the cursor position to the end of the line and switch to insert mode.

Normal Mode	Description	Command Mode	Description	Visual Mode	Description
i	Enter insert mode at the current cursor position.	:w	Save the file.	v	Enter visual mode to select text.
x	Delete the character under the cursor.	:q	Quit Vim.	y	Copy the selected text.
dd	Delete the current line.	:q!	Quit Vim without saving changes.	d	Delete the selected text.

Command		Description			
yy	Copy the current line.	:wq or :x:X	Save and quit Vim.	p	Paste the copied or deleted text.
p	Paste the copied or deleted text below the current line.	:s/old/new/g	Replace all occurrences of “old” with “new” in the file.		
u	Undo the last change.	:set nu or :set number	Display line numbers.		
Ctrl + R	Redo the last undo.				