**Linux Command:**

[**Linux Commands Cheat Sheet | GeeksforGeeks**](https://www.geeksforgeeks.org/linux-commands-cheat-sheet/#directory)

**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](https://www.geeksforgeeks.org/ls-command-in-linux/) | List files and directories. | * **-l**: Long format listing. * **-a**: Include hidden files hidden ones * **-h**: Human-readable file sizes. | * **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](https://www.geeksforgeeks.org/cd-command-in-linux-with-examples/) | Change directory. |  | * **cd /path/to/directory** changes the current directory to the specified path. |
| [pwd](https://www.geeksforgeeks.org/pwd-command-in-linux-with-examples/) | Print current working directory. |  | * **pwd** displays the current working directory. |
| [mkdir](https://www.geeksforgeeks.org/mkdir-command-in-linux-with-examples/) | Create a new directory. |  | * **mkdir my\_directory** creates a new directory named “my\_directory”. |
| [rm](https://www.geeksforgeeks.org/rm-command-linux-examples/) | Remove files and directories. | * **-r**: Remove directories recursively. * **-f**: Force removal without confirmation. | * **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 “file.txt” without confirmation. |
| [cp](https://www.geeksforgeeks.org/cp-command-linux-examples/) | Copy files and directories. | * **-r**: Copy directories recursively. | * **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**](https://www.geeksforgeeks.org/mv-command-linux-examples/) | Move/rename files and directories. |  | * **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](https://www.geeksforgeeks.org/touch-command-in-linux-with-examples/) | Create an empty file or update file timestamps. |  | * **touch file.txt**  creates an empty file named “file.txt”. |
| [cat](https://www.geeksforgeeks.org/cat-command-in-linux-with-examples/) | View the contents of a file. |  | * **cat file.txt**  displays the contents of the file “file.txt”. |
| [head](https://www.geeksforgeeks.org/head-command-linux-examples/) | Display the first few lines of a file. | * **-n**: Specify the number of lines to display. | * **head file.txt**  shows the first 10 lines of the file “file.txt”. * **head -n 5 file.txt**  displays the first 5 lines of the file “file.txt”. |
| [tail](https://www.geeksforgeeks.org/tail-command-linux-examples/) | Display the last few lines of a file. | * **-n**: Specify the number of lines to display. | * **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](https://www.geeksforgeeks.org/ln-command-in-linux-with-examples/) | Create links between files. | * **-s**: Create symbolic (soft) links. | * **ln -s source\_file link\_name**  creates a symbolic link named “link\_name” pointing to “source\_file”. |
| [find](https://www.geeksforgeeks.org/find-command-in-linux-with-examples/) | Search for files and directories. | * **-name**: Search by filename. * **-type**: Search by file type. | * **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](https://www.geeksforgeeks.org/chmod-command-linux/) | Change file permissions. | * **u**: User/owner permissions. * **g**: Group permissions. * **o**: Other permissions. * **+**: Add permissions. * **–**: Remove permissions. * **=**: Set permissions explicitly. | * **chmod u+rwx file.txt**  grants read, write, and execute permissions to the owner of the file. |
| [chown](https://www.geeksforgeeks.org/chown-command-in-linux-with-examples/) | Change file ownership. |  | * **chown user file.txt** changes the owner of “file.txt” to the specified user. |
| [chgrp](https://www.geeksforgeeks.org/chgrp-command-in-linux-with-examples/) | Change group ownership. |  | * **chgrp group file.txt** changes the group ownership of “file.txt” to the specified group. |
| [umask](https://www.geeksforgeeks.org/umask-command-in-linux-with-examples/) | Set default file permissions. |  | * **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](https://www.geeksforgeeks.org/tar-command-linux-examples/) | Create or extract archive files. | * **-c**: Create a new archive. * **-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. | * **tar -czvf archive.tar.gz files/**  creates a compressed tar archive named “archive.tar.gz” containing the files in the “files/” directory. |
| [gzip](https://www.geeksforgeeks.org/gzip-command-linux/) | Compress files. | * **-d**: Decompress files. | * **gzip file.txt** compresses the file “file.txt” and renames it as “file.txt.gz”. |
| [zip](https://www.geeksforgeeks.org/zip-command-in-linux-with-examples/) | Create compressed zip archives. | * **-r**: Recursively include directories. | * **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](https://www.geeksforgeeks.org/ps-command-in-linux-with-examples/) | Display running processes. | * **-aux**: Show all processes. | * **ps aux**  shows all running processes with detailed information. |
| [top](https://www.geeksforgeeks.org/top-command-in-linux-with-examples/) | Monitor system processes in real-time. |  | * **top** displays a dynamic view of system processes and their resource usage. |
| [kill](https://www.geeksforgeeks.org/kill-command-in-linux-with-examples/) | Terminate a process. | * **-9**: Forcefully kill a process. | * **kill PID**  terminates the process with the specified process ID. |
| [pkill](https://www.geeksforgeeks.org/kill-command-in-linux-with-examples/) | Terminate processes based on their name. |  | * **pkill process\_name**  terminates all processes with the specified name. |
| **pgrep** | List processes based on their name. |  | * **pgrep process\_name**  lists all processes with the specified name. |
| [grep](https://www.geeksforgeeks.org/grep-command-in-unixlinux/) | used to search for specific patterns or regular expressions in text files or streams and display matching lines. | * **-i**: Ignore case distinctions while searching. * **-v**: Invert the match, displaying non-matching lines. * **-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 the matching line. | * **grep -i “hello” file.txt** * **grep -v “error” file.txt** * **grep -r “pattern” directory/** * **grep -l “keyword” file.txt** * **grep -n “pattern” file.txt** In these examples we are extracting our desirec output from filename (file.txt) |

**5. System Information Commands**

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

| **sudCommand** | **Description** | **Options** | **Examples** |
| --- | --- | --- | --- |
| [uname](https://www.geeksforgeeks.org/uname-command-in-linux-with-examples/) | Print system information. | * **-a**: All system information. | * **uname -a** displays all system information. |
| [whoami](https://www.geeksforgeeks.org/whoami-command-linux-example/) | Display current username. |  | * **whoami**  shows the current username. |
| [df](https://www.geeksforgeeks.org/df-command-in-linux-with-examples/) | Show disk space usage. | * **-h**: Human-readable sizes. | * **df -h**  displays disk space usage in a human-readable format. |
| [du](https://www.geeksforgeeks.org/du-command-linux-examples/) | Estimate file and directory sizes. | * **-h**: Human-readable sizes. * **-s**: Display total size only. | * **du -sh directory/**  provides the total size of the specified directory. |
| [free](https://www.geeksforgeeks.org/free-command-linux-examples/) | Display memory usage information. | * **-h**: Human-readable sizes. | * **free -h**  displays memory usage in a human-readable format. |
| [uptime](https://www.geeksforgeeks.org/linux-uptime-command-with-examples/) | Show system uptime. |  | * **uptime** shows the current system uptime. |
| **lscpu** | Display CPU information. |  | * **lscpu**  provides detailed CPU information. |
| **lspci** | List PCI devices. |  | * **lspci** List PCI devices. |
| [lsusb](https://www.geeksforgeeks.org/lsusb-command-in-linux-with-examples/) | List USB devices. |  | * **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](https://www.geeksforgeeks.org/ifconfig-command-in-linux-with-examples/) | Display network interface information. | * **ifconfig**  shows the details of all network interfaces. |
| [ping](https://www.geeksforgeeks.org/ping-command-in-linux-with-examples/) | Send ICMP echo requests to a host. | * **ping google.com**  sends ICMP echo requests to “google.com” to check connectivity. |
| [netstat](https://www.geeksforgeeks.org/netstat-command-linux/) | Display network connections and statistics. | * **netstat -tuln**  shows all listening TCP and UDP connections. |
| **ss** | Display network socket information. | * **ss -tuln**  shows all listening TCP and UDP connections. |
| [ssh](https://www.geeksforgeeks.org/ssh-command-in-linux-with-examples/) | Securely connect to a remote server. | * **ssh user@hostname**  initiates an SSH connection to the specified hostname. |
| [scp](https://www.geeksforgeeks.org/scp-command-in-linux-with-examples/) | Securely copy files between hosts. | * **scp file.txt user@hostname:/path/to/destination**  securely copies “file.txt” to the specified remote host. |
| [wget](https://www.geeksforgeeks.org/wget-command-in-linux-unix/) | Download files from the web. | * **wget http://example.com/file.txt**  downloads “file.txt” from the specified URL. |
| [curl](https://www.geeksforgeeks.org/curl-command-in-linux-with-examples/) | Transfer data to or from a server. | * **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. |
| cmd > /dev/null | Discards the stdout of cmd by sending it to the null device. |
| cmd &> file | Every output of cmd is redirected to file. |
| cmd 1>&2 | stdout is redirected to the same place as stderr. |
| cmd >> file | 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. |

**10. Shortcuts Commands**

There are many shortcuts commands in Linux that can help you be more productive. Here are a few of the most common ones:

**10.1: Bash Shortcuts Commands:**

| **Navigation** | **Description** | **Editing** | **Description** | **History** | **Description** |
| --- | --- | --- | --- | --- | --- |
| **Ctrl + A** | Move to the beginning of the line. | **Ctrl + U** | Cut/delete from the cursor position to the beginning of the line. | **Ctrl + R** | Search command history (reverse search). |
| **Ctrl + E** | Move to the end of the line. | **Ctrl + K** | Cut/delete from the cursor position to the end of the line. | **Ctrl + G** | Escape from history search mode. |
| **Ctrl + B** | Move back one character. | **Ctrl + W** | Cut/delete the word before the cursor. | **Ctrl + P** | Go to the previous command in history. |
| **Ctrl + F** | Move forward one character. | **Ctrl + Y** | Paste the last cut text. | **Ctrl + N** | Go to the next command in history. |
| **Alt + B** | Move back one word | **Ctrl + L** | Clear the screen. | **Ctrl + C** | Terminate the current command. |
| **Alt + F** | Move forward one word. |  |  |  |  |

**10.2: Nano Shortcuts Commands:**

| **File Operations** | **Description** | **Navigation** | **Description** | **Editing** | **Description** | **Search and Replace** | **Description** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Ctrl + O** | Save the file. | **Ctrl + Y** | Scroll up one page. | **Ctrl + K** | Cut/delete from the cursor position to the end of the line. | **Ctrl + W** | Search for a string in the text. |
| **Ctrl + X** | Exit Nano (prompt to save if modified). | **Ctrl + V** | Scroll down one page. | **Ctrl + U** | Uncut/restore the last cut text. | **Alt + W** | Search and replace a string in the text. |
| **Ctrl + R** | Read a file into the current buffer. | **Alt + \** | Go to a specific line number. | **Ctrl + 6** | Mark a block of text for copying or cutting. | **Alt + R** | Repeat the last search. |
| **Ctrl + J** | Justify the current paragraph. | **Alt + ,** | Go to the beginning of the current line. | **Ctrl + K** | Cut/delete the marked block of text. |  |  |
|  |  | **Alt + .** | Go to the end of the current line. | **Alt + 6** | Copy the marked block of text. |  |  |

**10.3: 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. |
| **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. |

**10.4: Vim Shortcuts Commands:**

| **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. |
| **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. |  |  |  |  |