

Linux Commands Guide

Master the 10 Essential Linux Commands


1. ls - List Directory Contents

The `ls` command displays files and directories in the current or specified directory.

Syntax: `ls [options] [directory]`

Examples:

```
ls # List files in current directory
ls -l # List with detailed information
ls -a # List all files including hidden ones
ls -la /home/user # List all files with details
in /home/user
ls -lh # List with human-readable file sizes
```

 **Tip:** Use `ls -la` to see all files with permissions, ownership, and timestamps.

2. cd - Change Directory

The `cd` command changes your current working directory to navigate through the file system.

Syntax: `cd [directory]`

Examples :

```
cd /home/user # Go to specific directory
cd .. # Go to parent directory
cd ~ # Go to home directory
cd - # Go to previous directory
cd # Go to home directory (same as cd ~)
```

💡 **Tip:** Use `cd -` to quickly switch between two directories.

3. pwd - Print Working Directory

The `pwd` command shows the full path of your current working directory.

Syntax: `pwd`

Examples :

```
pwd # Show current directory path
Output: /home/user/documents
```

💡 **Tip:** Use `pwd` when you're lost in the file system to see where you are.


4. cat - Display File Contents

The **cat** command displays the contents of text files on the terminal.

Syntax: `cat [options] filename`

Examples:

```
cat file.txt # Display file contents
cat -n file.txt # Display with line numbers
cat file1.txt file2.txt # Display multiple
files
cat > newfile.txt # Create new file (type
content, Ctrl+D to save)
```

 **Tip:** Use `cat -n` to see line numbers, helpful for programming files.


5. cp - Copy Files and Directories

The **cp** command copies files or directories from one location to another.

Syntax: `cp [options] source destination`

Examples:

```
cp file.txt backup.txt # Copy file
cp -r folder1 folder2 # Copy directory
recursively
cp *.txt /home/user/documents # Copy all .txt
files
cp -i file.txt existing.txt # Copy with
confirmation prompt
```

 **Tip:** Use `cp -r` for directories and `cp -i` to avoid accidentally overwriting files.


6. mv - Move/Rename Files and Directories

The `mv` command moves files/directories or renames them.

Syntax: `mv [options] source destination`

Examples:

```
mv oldname.txt newname.txt # Rename file
mv file.txt /home/user/ # Move file to
directory
mv folder1 folder2 # Rename directory
mv *.txt documents/ # Move all .txt files
```

 **Tip:** Unlike `cp`, `mv` removes the original file/directory after moving.

7. chmod - Change File Permissions

The `chmod` command changes file and directory permissions.

Syntax: `chmod [permissions] filename`

Examples:

```
chmod 755 script.sh # rwxr-xr-x permissions
chmod +x file.txt # Add execute permission
chmod -w file.txt # Remove write permission
chmod u+r,g+w,o-x file.txt # Complex permission
changes
```

💡 **Tip:** Common permissions: 644 (rw-r--r--) for files, 755 (rwxr-xr-x) for directories.


8. chown - Change File Ownership

The **chown** command changes file and directory ownership.

Syntax: `chown [user:group] filename`

Examples:

```
chown user file.txt # Change owner to 'user'
chown user:group file.txt # Change owner and
group
chown -R user:group folder/ # Change ownership
recursively
chown :group file.txt # Change group only
```

 **Tip:** Use `sudo chown` when you need administrative privileges to change ownership.

9. grep - Search Text Patterns

The **grep** command searches for text patterns within files.

Syntax: `grep [options] pattern [file]`

Examples:

```
grep "error" logfile.txt # Search for "error"
in file
grep -r "function" . # Search recursively in
current directory
grep -i "warning" *.log # Case-insensitive
search
grep -n "TODO" code.py # Show line numbers with
matches
grep -v "debug" log.txt # Show lines that don't
match
```

💡 **Tip:** Use `grep -r` to search in all files within directories.

10. find - Search for Files and Directories

The `find` command searches for files and directories based on various criteria.

Syntax: `find [path] [expression]`

Examples:

```
find . -name "*.txt" # Find all .txt files
find /home -type d -name "project*" # Find
directories starting with "project"
find . -size +100M # Find files larger than
100MB
find . -mtime -7 # Find files modified in last
7 days
find . -name "*.log" -exec rm {} \; # Find and
delete .log files
```

💡 **Tip:** Use `find` with `-type f` for files only, `-type d` for directories only.

Quick Reference Summary

Command	Purpose	Common Usage
ls	List files	ls -la
cd	Change directory	cd /home/user
pwd	Show current path	pwd
cat	Display file contents	cat file.txt
cp	Copy files	cp -r folder1 folder2
mv	Move/rename files	mv old.txt new.txt
chmod	Change permissions	chmod 755 script.sh
chown	Change ownership	chown user:group file
grep	Search text patterns	grep -r "pattern" .
find	Search files/ directories	find . -name "*.txt"

Study Tips:

- Practice each command with different options
- Try combining commands using pipes (|)
- Use `man` command to see the manual for any command
- Start with simple examples and gradually try more complex ones
- Focus on understanding what each command does rather than memorizing syntax