Linux Commands Full Guide

ls - List Directory Contents

The `ls` command lists files and directories.

Syntax:

```
ls [OPTIONS] [PATH]
```

Common Options:

-l : Long listing format
-a : Include hidden files
-h : Human-readable sizes
-R : Recursive
-S : Sort by size
-t : Sort by modification time

Example:

```
ls -la
```

file - Identify File Type

The `file` command determines the file type.

Syntax:

```
file [OPTIONS] FILE
```

Example:

```
file image.png
```

less - View File Contents One Page at a Time

The `less` command lets you view file content page by page.

Syntax:

```
less [FILE]
```

Navigation:

Space: next page
b: previous page
q: quit

Linux Commands Full Guide

cp - Copy Files or Directories

The `cp` command copies files or directories.

Syntax:

```
   cp [OPTIONS] SOURCE DEST
```

Options:

-r : Recursive (for directories)
-i : Prompt before overwrite
-u : Only copy when source is newer

mv - Move or Rename Files

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

Syntax:

```
   mv [OPTIONS] SOURCE DEST
```

Options:

-i : Prompt before overwrite
-u : Move only if newer

mkdir - Make Directories

The `mkdir` command creates new directories.

Syntax:

```
   mkdir [OPTIONS] DIRECTORY
```

Options:

-p : Create parent directories as needed

rm - Remove Files or Directories

The `rm` command deletes files or directories.

Syntax:

```
   rm [OPTIONS] FILE
```

Options:

```
-r : Recursive (delete directories)
-f : Force deletion (ignore errors)
-i : Prompt before each deletion
```

man - Manual Pages

The `man` command shows help documentation for other commands.

Syntax:

```
man [COMMAND]
```

Example:

```
man ls
```

whatis - Quick Command Summary

The `whatis` command gives a one-line summary of a command.

Example:

```
whatis grep
```

alias - Create Shortcuts for Commands

The `alias` command creates command shortcuts.

Syntax:

```
alias name='command'
```

Example:

```
alias ll='ls -l'
```

env - Show Environment Variables

The `env` command displays environment variables.

Syntax:

```
env
```

Example:

```
   env | grep PATH
```

cut - Remove Sections from Each Line

The `cut` command removes sections of each line in a file.

Syntax:

```
   cut OPTION [FILE]
```

Examples:

```
   echo "ShravanBhise" | cut -c 8-12 -> Bhise
   cut -d ':' -f 1 /etc/passwd
```

paste - Merge Lines from Files

The `paste` command merges lines of files side by side.

Example:

```
   paste file1.txt file2.txt
```

head - View First Few Lines

The `head` command displays the first 10 lines by default.

Syntax:

```
   head [OPTIONS] FILE
```

Options:

```
   -n [N] : Show first N lines
```

tail - View Last Few Lines

The `tail` command displays the last 10 lines by default.

Syntax:

```
   tail [OPTIONS] FILE
```

Options:

```
    -n [N] : Show last N lines
    -f     : Follow file as it grows (useful for logs)
```

expand & unexpand - Tabs to Spaces and Back

`expand` converts tabs to spaces.
`unexpand` converts spaces to tabs.

Examples:

```
    expand file.txt
    unexpand file.txt
```

join - Join Two Files by a Common Field

`join` joins lines of two files on a common field.

Example:

```
    join file1.txt file2.txt
```

split - Split a File into Pieces

`split` splits large files into smaller pieces.

Example:

```
    split -l 50 bigfile.txt small_
```

sort - Sort Lines

The `sort` command sorts lines in a file.

Options:

```
    -r : Reverse order
    -n : Numerical sort
    -k N : Sort by column N
```

tr - Translate or Delete Characters

The `tr` command replaces or removes characters.

Examples:

```
    echo "hello" | tr a-z A-Z -> HELLO
    echo "123 456" | tr -d ' '
```

uniq - Filter Repeated Lines

The `uniq` command removes duplicate lines (adjacent only).

Options:

```
    -c : Count duplicates
    -d : Only duplicates
    -u : Only unique lines
```

Common: sort file.txt | uniq

wc & nl - Count and Number Lines

`wc` counts lines, words, characters.
Example:

```
    wc -l file.txt
```

`nl` numbers lines in a file.
Example:

```
    nl file.txt
```

grep - Search for Patterns

The `grep` command searches for patterns in files.

Options:

```
    -i : Ignore case
    -v : Invert match
    -n : Line numbers
    -c : Count matches
    -r : Recursive
    -w : Match whole word
```

Example:

```
    grep "text" file.txt
```

Linux Commands Full Guide

stdin, stdout, stderr - Standard Input/Output/Error

stdin: Standard input (usually from keyboard or input redirection)
stdout: Standard output (usually to screen, can be redirected)
stderr: Standard error output (for error messages)

Examples:

```
    command > out.txt       # stdout to file
    command 2> error.txt    # stderr to file
    command > out.txt 2>&1  # both stdout and stderr to file
```

pipe (|) and tee

A pipe (`|`) sends output from one command into another.

Example:

```
    ls | grep 'txt'
```

`tee` reads from stdin and writes to stdout and files.

Example:

```
    ls | tee list.txt
```