Let's dive deep into each of these commands, exploring their options, advanced usage, and examples.

# **1️⃣ mkdir (Make Directory)**

The mkdir command is used to create directories (folders).

## **Basic Syntax**

mkdir directory\_name

Creates a directory named directory\_name.

## **Common Options**

| **Option** | **Description** |
| --- | --- |
| -p | Create parent directories if they don’t exist |
| -v | Print a message for each created directory |
| -m mode | Set permissions for the directory |
| --help | Show help documentation |

### **Examples**

#### **1️⃣ Creating a simple directory**

mkdir myfolder

Creates myfolder in the current directory.

#### **2️⃣ Creating a nested directory**

mkdir -p projects/code/python

* -p: Ensures all parent directories exist.
* Creates projects/code/python.

#### **3️⃣ Creating a directory with specific permissions**

mkdir -m 755 secure\_folder

* -m 755: Sets read/write/execute for the owner, read/execute for others.

#### **4️⃣ Verbose output**

mkdir -v test\_dir

* -v: Prints a message confirming creation.

## **Practice Problems**

1. Create a directory named assignments.
2. Create a nested directory structure: work/docs/reports.
3. Create a directory secret with 700 permissions (owner full access, others no access).

# **2️⃣ ls (List Directory Contents)**

Lists files and directories in the current directory.

## **Basic Syntax**

ls [options] [directory]

## **Common Options**

| **Option** | **Description** |
| --- | --- |
| -a | Show hidden files (. prefix) |
| -l | Long format (permissions, owner, size, etc.) |
| -h | Human-readable sizes |
| -t | Sort by modification time |
| -r | Reverse order |
| -R | Recursive listing (subdirectories) |
| --color=auto | Colorize output based on file type |

### **Examples**

#### **1️⃣ List all files (including hidden)**

ls -a

Shows files starting with . (hidden files).

#### **2️⃣ Detailed listing with permissions**

ls -l

Outputs:

drwxr-xr-x 2 user user 4096 Feb 16 15:30 Documents

-rw-r--r-- 1 user user 1234 Feb 16 15:30 notes.txt

#### **3️⃣ Human-readable file sizes**

ls -lh

Shows file sizes like 4.0K, 2M, 1G instead of bytes.

#### **4️⃣ Sorting by modification time (latest first)**

ls -lt

#### **5️⃣ Recursive listing (including subdirectories)**

ls -R

## **Practice Problems**

1. List all files, including hidden ones.
2. List files sorted by last modification date.
3. Recursively list all files in Documents.

# **3️⃣ rmdir (Remove Empty Directories)**

Removes directories **only if they are empty**.

## **Basic Syntax**

rmdir directory\_name

## **Common Options**

| **Option** | **Description** |
| --- | --- |
| -p | Remove parent directories if empty |
| -v | Verbose mode (print messages) |

### **Examples**

#### **1️⃣ Removing an empty directory**

rmdir old\_folder

#### **2️⃣ Removing nested empty directories**

rmdir -p dir1/dir2/dir3

Removes dir3, dir2, and dir1 (if all are empty).

## **Practice Problems**

1. Create and remove an empty folder.
2. Remove a nested directory structure.

# **cp (Copy Files and Directories)**

Copies files and directories.

## **Basic Syntax**

cp [options] source destination

## **Common Options**

| **Option** | **Description** |
| --- | --- |
| -r | Copy directories recursively |
| -i | Interactive mode (prompt before overwriting) |
| -u | Copy only newer files |
| -v | Verbose mode |
| -p | Preserve file attributes (timestamps, permissions) |

### **Examples**

#### **1️⃣ Copying a file**

cp file.txt backup/

#### **2️⃣ Copying a directory**

cp -r project backup/

#### **3️⃣ Copy with overwrite confirmation**

cp -i file.txt backup/

#### **4️⃣ Copy only if newer**

cp -u new\_report.txt archive/

## **Practice Problems**

1. Copy file1.txt to backup/.
2. Copy projects/ into old\_projects/ while preserving timestamps.

# **5️⃣ mv (Move or Rename Files)**

Moves or renames files.

## **Basic Syntax**

mv [options] source destination

## **Common Options**

| **Option** | **Description** |
| --- | --- |
| -i | Prompt before overwriting |
| -u | Move only newer files |
| -v | Verbose mode |

### **Examples**

#### **1️⃣ Renaming a file**

mv oldname.txt newname.txt

#### **2️⃣ Moving a file**

mv report.txt Documents/

## **Practice Problems**

1. Rename data.csv to data\_backup.csv.
2. Move photos/ to backup/.

# **6️⃣ cat (Concatenate and Display Files)**

Displays file contents.

## **Basic Syntax**

cat [options] filename

## **Common Options**

| **Option** | **Description** |
| --- | --- |
| -n | Show line numbers |
| -b | Number only non-empty lines |
| -s | Squeeze multiple blank lines |

### **Examples**

#### **1️⃣ Viewing a file**

cat notes.txt

#### **2️⃣ Displaying with line numbers**

cat -n script.sh

#### **3️⃣ Combining multiple files**

cat file1.txt file2.txt > combined.txt

## **Practice Problems**

1. Display the contents of README.md.
2. Merge chapter1.txt and chapter2.txt into book.txt.

# **7️⃣ wc (Word Count)**

Counts lines, words, and characters in a file.

## **Basic Syntax**

wc [options] filename

## **Common Options**

| **Option** | **Description** |
| --- | --- |
| -l | Count lines |
| -w | Count words |
| -c | Count bytes |
| -m | Count characters |

### **Examples**

#### **1️⃣ Counting lines**

wc -l file.txt

#### **2️⃣ Counting words**

wc -w document.txt

#### **3️⃣ Counting characters**

wc -m article.txt

## **Practice Problems**

1. Count the words in story.txt.
2. Find the number of lines in data.log.

This should give you a **deep understanding** of these commands! Want more advanced use cases? 🚀