
Task 1 – Basic Listing & Counting

Write a script that:

1. Takes **one argument**: a directory path.
2. Prints:
    - Total number of items in that directory.
    - Number of files.
    - Number of sub-directories.
3. If the directory does not exist, print a clear error message.

Example usage (for you to test):

./task1.sh /some/directory

Task 2 – File Info Helper

Write a script that:

1. Asks the user to enter a **file name** (can be with or without path).
2. If the file exists, print:
    - File size in bytes.
    - Last modified time.
    - Permissions in symbolic form (e.g. `-rw-r--r--`).

3. If the file does not exist, print a suitable message.

Task 3 – Text File Backup

Write a script that:

1. Takes **one argument**: a directory path.
2. Creates a new folder inside it named `backup_YYYYMMDD` (use the current date).
3. Copies all `.txt` files from the given directory into this backup folder.
4. Prints how many files were copied.

Task 4 – Top N Largest Files

Write a script that:

1. Takes **two arguments**:
    - A directory path.
    - A number `N`.
2. Finds the **N largest regular files** inside that directory (non-recursive).
3. Prints their sizes and names in **descending** order of size.
4. If `N` is missing or not a number, handle the error.

Task 5 – Simple Log Search

Write a script that:

1. Takes **two arguments**:

    - A directory path.
    - A search word.
2. Searches all `.log` files inside that directory (non-recursive) for lines that contain the word.
3. For every `.log` file that has at least one match, print the file name and number of matching lines.
4. If no `.log` files are found, print a message.

Task 6 – Hidden Files Report

Write a script that:

1. Looks in the **home directory** of the current user.
2. Lists all **hidden files** (not directories) directly inside it (non-recursive).
3. Prints:
    - Total number of hidden files.
    - A list of their names.

Task 7 – Simple Change Tracker

Write a script that:

1. Takes **one argument**: a directory path.
2. Creates (or updates) a file named `.snapshot.txt` inside that directory containing the list of files (non-recursive, one per line).
3. On the **next run**, compares the current list of files with `.snapshot.txt` and prints:
    - Files that are **new**.

    - Files that are **missing** compared to the last run.

4. After printing, update `.snapshot.txt` with the new list.

Task 8 – Basic Symlink Checker

Write a script that:

1. Takes **one argument**: a directory path.
2. Lists all **symbolic links** directly inside that directory.
3. For each symbolic link, prints whether it points to a valid target or a missing target.

Task 9 – Space Threshold Action

Write a script that:

1. Takes **one argument**: a directory path.
2. Calculates the total size (in MB) of that directory (non-recursive).
3. If the size is greater than a fixed threshold you choose (for example, 100 MB):
    - Print a warning message.
    - Create a file named `size_warning.txt` inside that directory containing the date and the size.
4. If the size is less than or equal to the threshold, print that everything is OK.

Task 10 – Simple File Type Classifier (By Extension)

Write a script that:

1. Takes **one argument**: a directory path.
2. For each item in that directory (non-recursive):
    - If it is a file, check its extension (e.g. `.txt`, `.sh`, `.jpg`).
    - Count how many files there are of each extension.
3. At the end, print a summary like:
    - `txt: 5 files`
    - `sh: 3 files`
    - `no extension: 2 files`