Placement Empowerment Program

Cloud Computing and DevOps Centre

Day 08 – File Organizer by Type

Automatically sort files in a directory into subfolders based on their file type or extension.

Name: Hema S

Department: ECE

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY

AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

In this Proof of Concept (PoC), I automated the process of organizing files in a directory based on their file types or extensions using a Bash script.

The goal of this PoC was to reduce clutter in folders (like Downloads or project workspaces) by programmatically sorting files into categorized subfolders. Each file is moved into a folder named after its extension **(e.g., .pdf files into a pdf/ folder, .jpg files into a jpg/ folder, etc.).** This automation not only improves productivity and efficiency but also serves as a practical example of how simple Linux scripting can solve everyday problems with minimal effort.

Overview

This PoC demonstrates the creation of a Bash script that automatically organizes files in a directory by sorting them into subfolders based on their file extensions.
The script scans through all regular files in a given directory, identifies their file types by extension, and moves each file into a corresponding folder **(e.g., pdf/, jpg/, txt/, etc.)**. If the folder doesn't exist, it is created dynamically during execution.
This automation significantly improves file organization, especially in cluttered directories like Downloads, shared folders, or code projects. It's a practical example of how Linux shell scripting can be used for efficient system management and everyday automation tasks.

Key steps in this PoC:

1. **Create a Working Directory (Optional for Testing)**
Set up a test folder with sample files of different types using **mkdir** and touch.

2. Write the Bash Script
A script **(organize_by_type.sh)** was created using **nano**, containing

logic to:

✅Loop through files in the target directory

✅Extract file extensions

✅Convert extensions to lowercase for consistency

✅Create a folder for each extension (if not already existing)

✅Move the file into the appropriate folder

3.Make the Script Executable

The script was made executable using:

```
bash
 CopyEdit
chmod +x organize_by_type.sh
```

4. Run the Script

Executed the script with:

```
bash
 CopyEdit
./organize_by_type.sh <target-directory>
```

If no directory is passed, it organizes files in the current folder.

5. **Verify the Output**

Checked the directory to confirm that all files were sorted into subfolders like **txt/, pdf/, jpg/**, etc.

Objectives :

✅ Automate file organization by sorting files into subfolders based on their file extensions.

✅ Improve folder cleanliness and navigation, especially in directories like Downloads, project folders, or shared workspaces.

✅ Gain hands-on experience with Bash scripting and Linux file-handling commands.

✅ Practice conditional logic and string operations in shell scripts (e.g., extracting file extensions, checking file types).

✅ Create a reusable tool that can be executed on any directory to instantly organize its contents.

✅ Demonstrate real-world use of Linux automation to simplify daily system maintenance tasks.

Importance :

✅ Saves time and effort by eliminating the need to manually organize files in large or cluttered directories like Downloads or project folders.

✅ Enhances productivity by maintaining a clean and well-structured workspace, making it easier to locate specific files quickly.

✅ Reduces human error, especially in shared environments where improper file placements can lead to confusion or data loss.

✅ Demonstrates practical Linux scripting skills, showcasing how automation can solve real-world problems with minimal code.

✅ Reusable and customizable, making it a handy utility script for both personal and professional use.

✅ Promotes better digital hygiene, encouraging organized file storage and reducing chaos in frequently used folders.

Step-by-Step Overview

Step 1: Launch Terminal

Open the terminal on your Linux system to begin scripting and testing.

Step 2: Create a Test Directory (Optional)

Create a folder to simulate an unorganized environment:

```
hemas@Hema:/mnt/c/Users/hemas$ mkdir ~/file-organizer
hemas@Hema:/mnt/c/Users/hemas$ cd ~/file-organizer
```

Step 3: Add Sample Files

Create multiple files with different extensions to test the organizer:

```
hemas@Hema:~/file-organizer$ touch file1.txt file2.txt image1.jpg vid
eo1.mp4 doc1.pdf script.sh
```

Use ls to confirm:

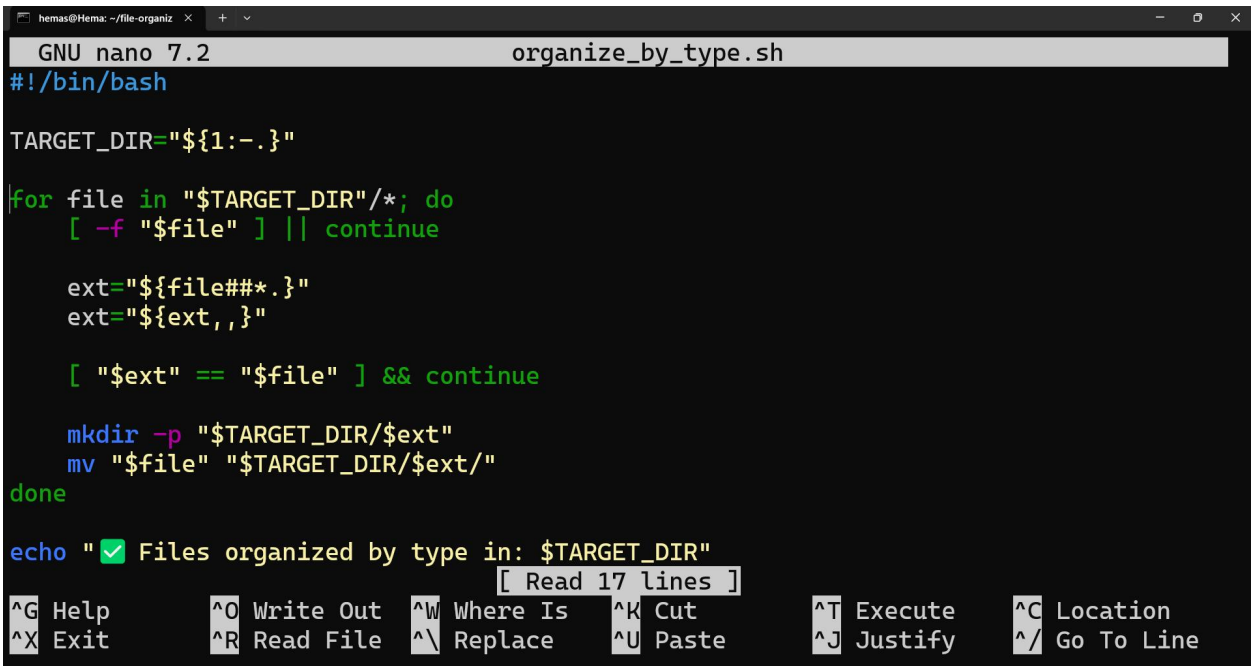
```
hemas@Hema:~/file-organizer$ ls
doc1.pdf  file1.txt  file2.txt  image1.jpg  script.sh  video1.mp4
```

Step 4: Create the Organizer Script

Open a new Bash Script file:

```
hemas@Hema:~/file-organizer$ nano organize_by_type.sh
```

Paste the following code into the editor:



✅Save with Ctrl + O, press Enter
✅ Exit with Ctrl + X

Step 5: Make the Script Executable

```
hemas@Hema:~/file-organizer$ chmod +x organize_by_type.sh
```

Step 6: Run the Script

To organize the current folder:

```
hemas@Hema:~/file-organizer$ ./organize_by_type.sh
✅ Files organized by type in: .
```

Step 7: Check the Output

Use ls to confirm that your files are now in subfolders:

```
hemas@Hema:~/file-organizer$ ls
jpg  mp4  pdf  sh  txt
```

Check inside one:

```
hemas@Hema:~/file-organizer$ ls txt
file1.txt  file2.txt
```

Step 8: Use It Anywhere

You can now use this script on any messy directory by simply pointing to it:

```
hemas@Hema:~/file-organizer$ ./organize_by_type.sh ~/Downloads
✅ Files organized by type in: /home/hemas/Downloads
```

Outcomes:

✅ Successfully developed a Bash script to organize files by their extensions.

✅ Practiced file handling, string manipulation, and conditional logic in shell scripting.

✅ Demonstrated the ability to automate repetitive tasks using simple scripting techniques.

✓ Reduced manual file organization effort by programmatically sorting files into subfolders like **pdf/, jpg/, txt/, etc.**

✓ Created a reusable and portable tool that can organize files in any directory when executed.

✓ Improved productivity and directory cleanliness—especially useful in folders like Downloads, shared workspaces, and project folders.

✓ Strengthened understanding of Linux commands such as **mkdir, mv, chmod, nano, and path handling.**