ECE2800J Exercise 1

Introduction

Welcome to ECE2800J coding exercise 1! The exercises aim to consolidate what you've learned in the lectures. They will not be hard, so take it easy and enjoy coding! And don't get frustrated when you make mistakes. It's normal to get stuck and have multiple attempts even for the best programmers. Anyway, let's get started!

Background

In this exercise, you will write a bash script to practice and enhance your skill in basic linux commands. This exercise requires you to write a bash script that automates the creation and management of a directory named `MyJournal` and its related files and subdirectories, including content manipulation and organizational tasks.

> A bash script is a file containing a series of commands that are executed as if they were entered on the command line. The execution sequence is the same as the order of the commands in the file.

Your Tasks

Write a Bash script named `ex1.sh` that performs the following operations **in order**. All operations must use **relative paths** and be self-contained within the script.

1. Create the Journal Directory Structure

- Create a directory called `MyJournal`.
- Enter `MyJournal` and create two subdirectories:
  - `entries`: to store daily journal entries
  - `archive`: to store archived or backed-up files

2. Initialize the Configuration File

- In the `MyJournal` root directory, create a file named `config.txt`.
- Use `echo` and output redirection (`>`) to add the following content into `config.txt`:

```
Default Format: YYYY_MM_DD_entry.txt
```

3. Create Initial Journal Entries

- Navigate into the `entries` directory.
- Use `touch` to create two empty journal files:
- `2025_09_01_entry.txt`
- `2025_09_02_entry.txt`
- Use `echo` and output redirection (`>`) to add a short message to each file:

```
Today is a productive day.
```

4. Backup the Entries

- Go back to the `MyJournal` directory.
- Copy the entire `entries` directory into `archive` as a backup and rename it as `backup_entries`, so the path becomes:

```
archive/backup_entries/
```

5. Update and Compare a Journal Entry

- Navigate back to the `entries` directory.
- Use `echo` command to append the following line to `2025_09_02_entry.txt` using `>>`:

```
Updated with evening thoughts.
```

- Use the `diff` command to compare
  - The original version: `../archive/backup_entries/2025_09_02_entry.txt`
  - The updated version: `2025_09_02_entry.txt`

  and save the output of `diff` into a file called `diff_result.txt` in the `MyJournal` root directory.

6. Rename and Move an Entry

- Rename `2025_09_01_entry.txt` to `old_diary.txt`.
- Move `old_diary.txt` from `entries` to the `archive` directory.

7. Cleanup and Documentation

- Navigate to `MyJournal` root directory.
- Delete the `archive/backup_entries` directory and all its contents.
- In the `MyJournal` root directory:
  - Create a hidden configuration file named `.journal_settings` and use `echo` to add the following content into it:

    ```
    auto_backup=true
    ```

  - Run `ls -laR` to recursively list all files, and save the output to `directory_report.txt`.

Requirements

- The first line of `ex1.sh` must be:

```
#!/bin/bash
```

- Do not use absolute paths (e.g., `/home/...`). Use `./`, `../`, or no prefix.
- **Never** run `ex1.sh` in `/`, `/root`, or system directories.
- Ensure the script is executable and well-formatted.

Testing

To execute the script, you need to make it executable by running the following commands in your terminal:

```
chmod +x ex1.sh # make the script executable
./ex1.sh # execute the script
```

If you encounter any errors, a dumb way to debug is to execute the commands in the script manually one by one and check the corresponding outputs and directory structures.

Submission

Submit your `ex1.sh` to canvas directly. See due date on canvas.