Lab 1: Directory and File Creation Script
Name: Dylan Winter
Date: January 27, 2025

****Introduction****
This project involves creating a shell script in Linux to automate the creation of directories and files with specific naming conventions and contents. The script is designed to enhance familiarity with basic shell scripting, directory and file manipulation, and essential Linux commands. By completing this project, users gain experience in automating repetitive tasks and working with the Linux command line.

****How the Script Works****
The shell script automates the following tasks:

1. Create the Main Directory
   - The script uses the `date` command to generate the current date and time in the format `YYYY-MM-DD_HH-MM-SS`.
   - This string is used as the name of the main directory, which is created using the `mkdir` command.

2. Create Subdirectories
   - A loop iterates ten times to create subdirectories named `file101` to `file110` inside the main directory.

3. Generate .txt Files with Programming Languages
   - Inside each subdirectory, a `.txt` file named `tuser50X.txt` is created, where `X` corresponds to the subdirectory number (e.g., `tuser50101.txt` in `file101`).
   - Each `.txt` file contains a unique programming language, such as Python, Java, or C++.

****Example Output Structure****
```
2025-01-27_12-34-56/
        file101/
        tuser50101.txt (contains "Python")
        file102/
        tuser50102.txt (contains "Java")
        file103/
        tuser50103.txt (contains "C++")
        ...
```

Linux Commands Overview

mkdir

- Description: Creates directories.
- Usage: mkdir directory_name
- Example in Script: Creates the main directory and subdirectories.

echo

- Description: Outputs text to the terminal or writes text to a file.
- Usage: echo "text" > file_name
- Example in Script: Writes programming language names into .txt files.

chmod

- Description: Changes file permissions.
- Usage: chmod +x script_name
- Example in Script: Makes the shell script executable.

date

- Description: Fetches the current date and time.
- Usage: date +"%Y-%m-%d_%H-%M-%S"
- Example in Script: Generates the name for the main directory.

cp

- Description: Copies files or directories.
- Usage: cp source_file destination_file
- Example: cp file1.txt file2.txt creates a copy of file1.txt named file2.txt.

ps

- Description: Displays information about running processes.
- Usage: ps aux
- Example: ps aux shows all currently running processes along with details such as user, CPU usage, and memory usage.

ls

- Description: Lists files and directories in a directory.
- Usage: ls -l
- Example: ls -l displays detailed information about each file in the current directory, including permissions, size, and modification date.

mv

- Description: Moves or renames files or directories.
- Usage: mv source_file destination_file
- Example: mv file1.txt file2.txt renames file1.txt to file2.txt.

rm

- Description: Deletes files or directories.
- Usage: rm file_name
- Example: rm file1.txt deletes the file file1.txt.

rmdir

- Description: Removes empty directories.
- Usage: rmdir directory_name
- Example: rmdir my_folder removes the directory my_folder if it is empty.

more

- Description: Displays the content of a file one screen at a time.
- Usage: more file_name
- Example: more file1.txt displays the content of file1.txt, pausing after each screenful of content.

time

- Description: Measures the execution time of a command.
- Usage: time command
- Example: time ./script.sh measures how long it takes to execute the script script.sh.

kill

- Description: Terminates a process by its PID (Process ID).
- Usage: kill PID
- Example: kill 1234 terminates the process with PID 1234.

history

- Description: Displays the history of commands executed in the terminal.
- Usage: history
- Example: history | grep mkdir shows all previous mkdir commands executed.

chown

- Description: Changes the ownership of a file or directory.
- Usage: chown user:group file_name
- Example: chown john:developers file1.txt changes the owner of file1.txt to john and the group to developers.

****Screenshots****

1. Folder Structure:
   - A screenshot showing the created main directory (`2025-01-27_XX-XX-XX`), subdirectories (`file101` to `file110`), and `.txt` files.

2. File Contents:
   - A screenshot of the contents of one `.txt` file (e.g., `tuser50101.txt` containing "Python").

GitHub Link
https://github.com/dylanwinter200/linux-file-script

Conclusion
This project demonstrates the power and utility of Linux shell scripting to automate repetitive tasks such as creating directories and files with specific contents. By using commands like `mkdir`, `echo`, `chmod`, and `date`, this script effectively simplifies directory and file management. The project provides valuable experience in scripting and the Linux command line, contributing to practical skills in automation and system administration.