Operating Systems: cli

1 Introduction

The Command Line Interface (CLI) is a text-based interface that allows users to interact with the operating system using commands. It provides powerful tools for file management, process control, networking, and system administration. The CLI is accessed through a terminal and enables automation through shell scripting. Common shells include Bash, Zsh, and Fish, with commands like ls, cd, grep, and chmod being frequently used. The CLI is essential for advanced users and system administrators due to its efficiency, flexibility, and control over the system.

2 Background

In this exercise, we combine scripting along with cli commands to list out the files present in a given directory.

3 Programs

3.1 file.sh

This Bash script prompts the user for a directory path, checks if it exists, and then counts the number of files within it, displaying the result.

3.1.1 Detailed Explanation

1. **Shebang Line:** *(#!/bin/bash)* indicates that the script should be executed using the Bash shell.
2. **Prompt User for Directory Path:** *(read -p "Enter the directory path: " DIR)* asks the user to enter a directory path and stores the input in the DIR variable.
3. **Check if the Directory Exists:** −d "$DIR" checks if the specified directory exists. The ! negates the condition, meaning if the directory does NOT exist, then it Prints an error message and exits the script with a status of 1 (indicating an error).

4. **Count the Number of Files:**
    - *ls −1q "$DIR"*: Lists all files and directories in the given directory (-1 ensures one entry per line). -q ensures special characters are handled safely.
    - *wc −l*: Counts the number of lines (i.e., number of files and directories).
5. **Display the File Count:** *(if ["$FILE_COUNT" −gt 0]; then ... fi)* If FILE_COUNT is greater than 0, it prints the number of files. Otherwise, it prints "No files found in $DIR."

3.1.2 Conclusion

In this way, cli commands can be combined with bash scripting to achieve complex operations.

4 Takeaways

Please try to write bash scripts to do tasks like provide CPU usage, kill a running process, print the disk usage, check internet speed, etc,.