Programming Assignment 5
Shell Scripting for System Administration

Assignment spec may change depending upon discussion in the class, this is a working doc

Release Time	Due Date
November 16, 2023	November 22, 2023

Objectives

The purpose of this assignment is to enhance your skills in Unix/Linux shell scripting. You will create a series of shell scripts that demonstrate your ability to automate system administration tasks. This assignment will test your understanding of shell commands, scripting syntax, file handling, process management, and basic system administration. These skills will also enable you to incorporate HPC in your software development.

Prerequisites:

- Basic understanding of Linux/Unix operating system.
- Familiarity with command-line interface and basic commands.
- Access to a Linux or Unix-like environment for testing scripts.

Tasks:

1. **User Information Script**
   - Write a script `userinfo.sh` that accepts a username as an argument and displays the following information about the user:
     - User's full name
     - Home directory
     - Shell type
   - If the user does not exist, the script should display an appropriate message.
2. **System Health Check Script**
   - Create a script `healthcheck.sh` that reports the following system information:
     - Current date and time
     - System uptime
     - Total number of users currently logged in
     - Memory usage (free/used memory)
     - Disk usage
   - The script should format the output in a readable and organized manner.
3. **Directory Backup Script**
   - Write a script `backupdir.sh` that takes a directory path as input and creates a compressed backup of the entire directory in a specified backup location.

- The backup file name should include the current date.
- If the input is not a directory, the script should display an error message.

4. **Batch File Rename**
   - Develop a script `rename.sh` that renames all files in a given directory by adding a prefix provided as an argument.
   - The script should handle cases where no prefix is provided and display usage in such cases.
5. **Process Monitor**
   - Script `processmonitor.sh` should monitor a specific process (process name given as an argument).
   - If the process is not running, the script should start the process and log this event.
   - If the process is running, the script should log its current memory and CPU usage.

Deliverables:

1. The scripts `userinfo.sh`, `healthcheck.sh`, `backupdir.sh`, `rename.sh`, `processmonitor.sh`.
2. A `README.md` file documenting:
   - How to run each script.
   - Example usage and expected output for each script.
   - Any assumptions or additional features implemented.

Evaluation Criteria:

- Functionality: Scripts should work as specified in the requirements.
- Robustness: Scripts should handle errors and edge cases gracefully.
- Code Quality: Scripts should be well-organized, commented, and follow best practices.
- Documentation: The README file should be clear, concise, and accurate.

Note:

- Test your scripts in a safe and controlled environment. Do not run scripts on critical systems without proper testing and permissions.
- Ensure your scripts are executable (`chmod +x scriptname.sh`) before submission.

Congratulations! You're well on your way to becoming a system administrator! 🔍

References

- Gen-AI tools
- TutorialsPoint - https://www.tutorialspoint.com/unix/shell_scripting.htm
- Coursera - https://www.coursera.org/articles/what-is-shell-scripting#

If you have creative ideas for extensions or making it more interesting, run them by the course staff, and we'd be happy to give you guidance!

Design Requirements

Code Documentation

For this assignment, you must properly document your code and use good software development practices.

Github

Use github to store your repository. Use good revision-control-system practices as you develop various pieces of the search engine.

Testing

Make sure you test your application with several different values capturing different cases, to make sure it works.

Assignment Submission

- Generate a .zip file that bundles all your files, including:
  - Readme file
  - Source code files
  - any input or output files
  - sample test runs
- Don't forget to follow the naming convention specified for submitting assignments
- You will also show execution of your application to grader / instructor. They may give you a test case or two on the spot.