ELEC377 - Operating System

Lab 4 - Shell Scripting, ps.sh

Design Document

Bradley Stephen & Ben Wyand

Problem Statement

The goal of this lab is to write a shell script (ps.sh) that mimics the behavior of the ps command to display information about currently running processes in Linux. The script should list each process's PID (Process ID), user (owner of the process), group, RSS (Resident Set Size, or memory usage in KB), and the command associated with the process. The script should support command-line flags to allow users to specify which columns they want in the output:

1. -rss: Include RSS (memory usage) in the output.
2. -comm: Include the command name in the output.
3. -command: Include the full command line used to start the process.
4. -group: Include the group name of the process.

The script must avoid allowing both -comm and -command at the same time, as these options are mutually exclusive. It should also handle cases where the specified flags do not match any available information (e.g., if a process does not have a command line).

The script should use the /proc directory to gather information, specifically:

- /proc/[PID]/status: For extracting the PID, UID, GID, RSS, and command name.
- /proc/[PID]/cmdline: For the full command line associated with the process.

Solution Explanation

1. **Flag Parsing**: The script first processes the command-line flags and sets variables (showRSS, showCommand, showComm, showGroup) based on the user's selection. It checks for invalid flags and handles errors when both -comm and -command are provided.
2. **Data Collection**:

- The script iterates through directories in /proc with numeric names, each representing a process.
- For each process, the script extracts the necessary information:
  - **PID**: Extracted from /proc/[PID]/status.
  - **User and Group Names**: Retrieved using getent based on the UID and GID from /proc/[PID]/status.
  - **RSS**: Retrieved from the VmRSS entry in /proc/[PID]/status and set to zero if not available.
  - **Command**: Retrieved based on the flags; if -command is specified, the command line is extracted from /proc/[PID]/cmdline. If -comm is specified, the short name is extracted from the Name field in /proc/[PID]/status.

3. **Output Formatting and Sorting**: The script formats the output based on the chosen flags and writes it to a temporary file. After collecting data, the file is sorted by PID, and the results are displayed with aligned columns.
4. **Temporary File Management**: A temporary file is used to store the sorted output, which is deleted after the program completes.

This approach ensures that the script displays the information dynamically and accurately simulates the functionality of the ps command with custom options for selecting columns.