ELEC377 - Operating System

Lab 4 - Shell Scripting, ps.sh

Testing Document

Bradley Stephen & Ben Wyand

Test Data

1. **System Output (Command)**:

ps -eo pid,user,group,rss,comm > t1system.txt

This command captures the output of the ps command with pid, user, group, rss, and comm columns and saves it to t1system.txt.

2. **Script Output (Command)**:

./ps.sh -group -rss -comm > t1lab.txt

This command runs the custom ps.sh script with -group, -rss, and -comm flags, which corresponds to the same data columns as the system ps command. The output is saved to t1lab.txt.

Differences in Outputs

1. **Process Availability**:
    - The system ps command and the custom ps.sh script may show slight differences in the list of processes due to timing. Processes are created and terminated frequently, and a process might appear in one output but not the other if it starts or ends between the two command executions.
    - Short-lived processes, such as certain system or user commands that run and exit quickly, are more likely to be missing from one output but present in the other.
2. **Kernel vs. User Processes**:
    - The ps command might handle kernel processes differently compared to ps.sh. The system ps command is optimized to display information across both kernel and user processes seamlessly, while the ps.sh script depends

on the availability of /proc/[PID] files, which can vary in how kernel threads or processes expose their data.

   - Some kernel threads may have restricted or empty entries in fields like rss or comm, and the system ps may process these fields more accurately or skip displaying them, whereas ps.sh may attempt to show these entries with default values.

3. **Empty or Missing Data Fields**:
   - The ps.sh script sets default values if certain fields are missing or empty (e.g., 0 for RSS if not available), while the system ps command may handle missing fields differently or omit certain details for kernel threads. For example, kernel threads often have no memory usage (RSS) or command line, and the system ps command might leave such fields blank or represent them in a specific way.
   - Commands with no associated cmdline entry are handled differently: ps.sh falls back to the process name in the Name field, whereas the system ps command may handle this case with a specific notation or leave it blank.

4. **User and Group Resolution**:
   - Differences in user and group resolution may occur if the script or ps command cannot resolve a specific UID or GID to a username or group name. In some cases, this could result in discrepancies in the output where ps.sh displays numeric IDs if it cannot resolve them to names, whereas ps may show a name or handle this differently based on its internal resolution mechanisms.

5. **Command Truncation**:
   - For the comm field, ps may truncate or display shorter versions of commands, especially for longer names, while ps.sh might display the full Name field content from /proc/[PID]/status. The system ps command has predefined ways of truncating or shortening commands to fit within display constraints, which might differ slightly from how ps.sh handles the Name entry.

In summary, the primary differences between the outputs of ps and ps.sh are due to process timing, handling of kernel versus user processes, treatment of missing or empty data fields, and minor variations in user and group name resolution. These factors result in slight variations in the information displayed by each command, even though they generally provide similar content for active processes.