Test Cases 377 Lab 4

Gia Lee, Valerie So, William Robson

Outline

Assumptions: This test document is to test the functionality to list running processes using /proc interface using shell programming from scratch.

Testing the Shell Script:

For a test to be successful, the result must include:

1. The correct output given an input flag.
2. All the running processes in the system in order relative to the UID.
    a. In other test cases where the output in based on gid, uid and pid. The output will be relative to the first column of numbers.
3. A temporary file is generated but deleted once the process completes.
4. The process must terminate when it is complete.

For this test, each source file will have values labeled <Test case number> <lab or system>.txt

Lab test cases are testing the outputs given certain flags.

System test cases are testing the outputs given pid, gid, uid, user and group.

T1lab and T1 system are the first default test cases given in the lab instructions.

For example: t2lab.txt which is test case 2 testing the outputs given the Comm flag as its argument.

t2system.txt is test case 2 testing the outputs given pid and group as the arguments.

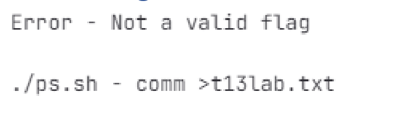
Lab Test Cases

In all lab test cases, there are two columns that will always be present. They are UID and USER. The third column will depend on which flag was indicated when running the script. The lists will all be sorted numerically according to the UID in the first column.

Ex: ./ps.sh -comm will have three columns, UID USER COMM

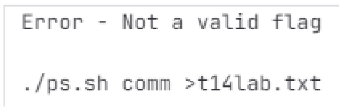
There are 14 lab test cases to showcase the outputs of running the "./ps.sh" command with up to 2 different flags.

Invalid Flag



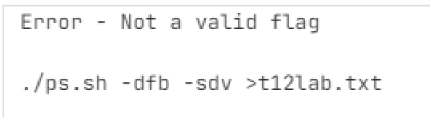
```
Error - Not a valid flag

./ps.sh - comm >t13lab.txt
```



```
Error - Not a valid flag

./ps.sh comm >t14lab.txt
```

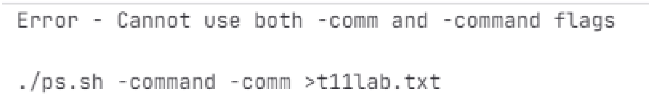


```
Error - Not a valid flag

./ps.sh -dfb -sdv >t12lab.txt
```

All the above test cases show that if the command has an invalid flag or does not follow the syntax, it will output an error indicating that the command is not a valid flag that it recognizes.

Running both -comm and -command flag together

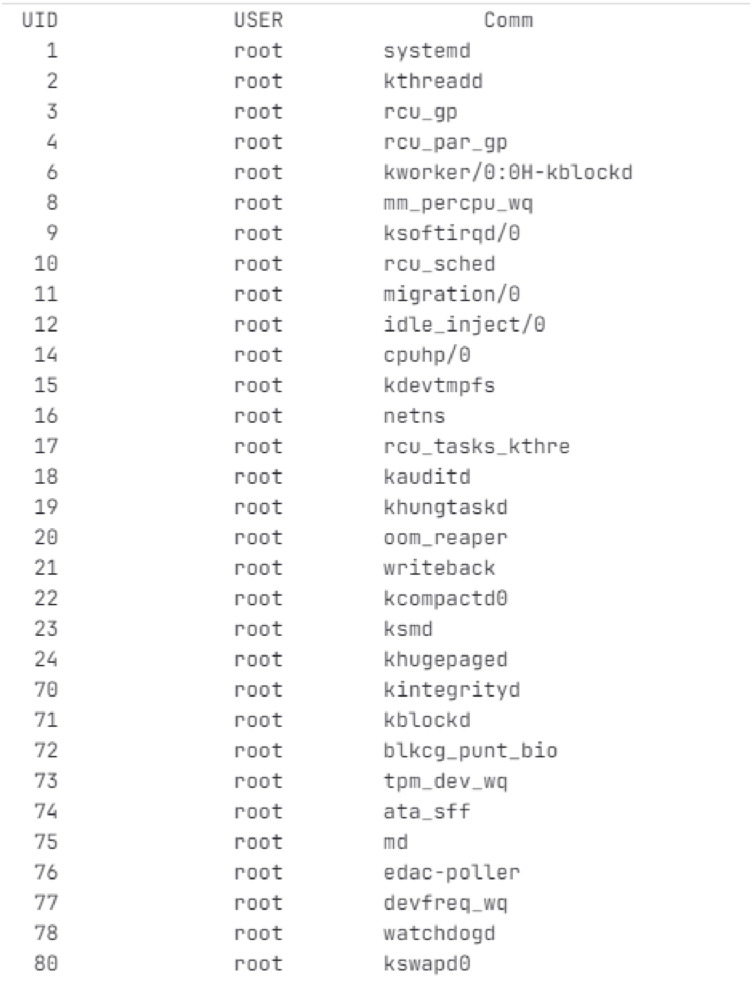


```
 Error - Cannot use both -comm and -command flags

 ./ps.sh -command -comm >t11lab.txt
```

The script reports the correct error if the user gives both -comm and -command flags in the same command line.

Valid Outputs



```
 UID                USER                Comm
   1                root        systemd
   2                root        kthreadd
   3                root        rcu_gp
   4                root        rcu_par_gp
   6                root        kworker/0:0H-kblockd
   8                root        mm_percpu_wq
   9                root        ksoftirqd/0
  10                root        rcu_sched
  11                root        migration/0
  12                root        idle_inject/0
  14                root        cpuhp/0
  15                root        kdevtmpfs
  16                root        netns
  17                root        rcu_tasks_kthre
  18                root        kauditd
  19                root        khungtaskd
  20                root        oom_reaper
  21                root        writeback
  22                root        kcompactd0
  23                root        ksmd
  24                root        khugepaged
  70                root        kintegrityd
  71                root        kblockd
  72                root        blkcg_punt_bio
  73                root        tpm_dev_wq
  74                root        ata_sff
  75                root        md
  76                root        edac-poller
  77                root        devfreq_wq
  78                root        watchdogd
  80                root        kswapd0
```

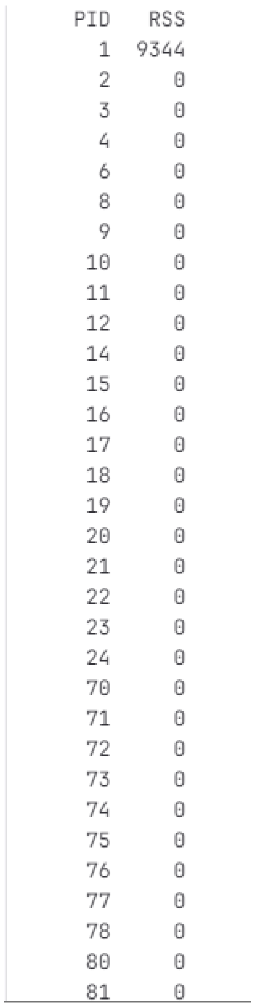
Example of a valid output where there are three columns including UID, USER and the flag that was inputted as the argument.

System Test Cases

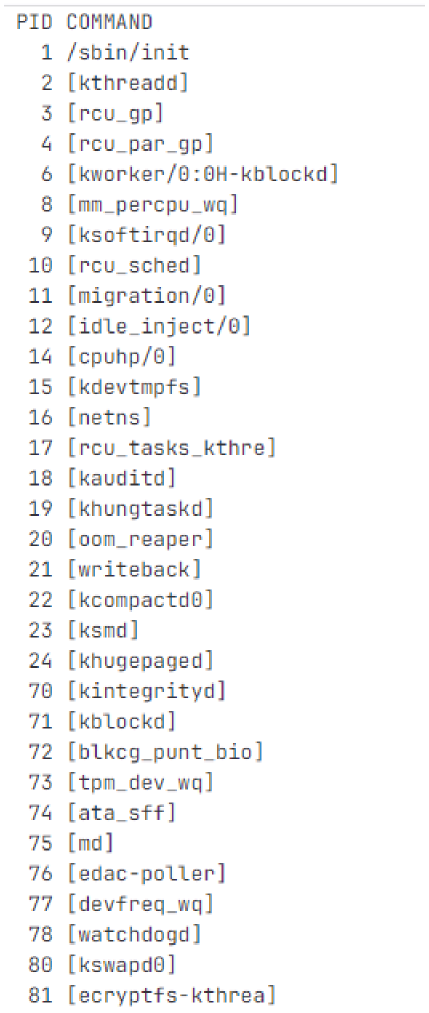
In all system test cases, there are two columns that will always be present depending on the arguments. The lists will all be sorted numerically according to the first column.

Ex: ps -eo pid, user will have two columns, PID USER. The list will be sorted numerically by PID in this case

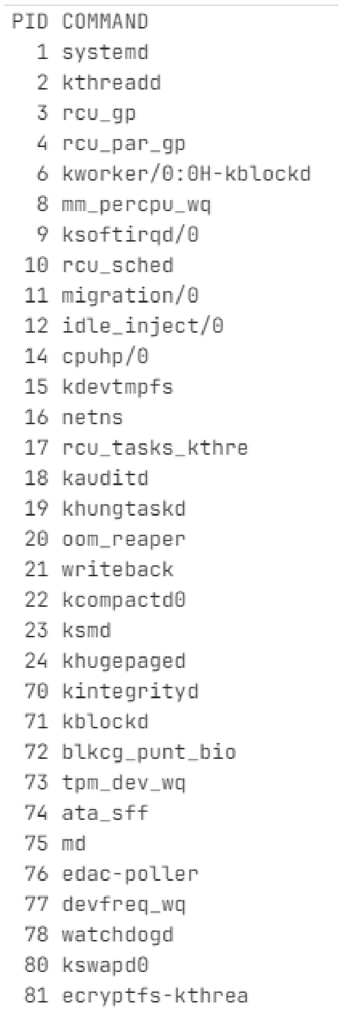
There are 17 system test cases to show that the output only displays the information the user asked for.



```
PID   RSS
  1  9344
  2     0
  3     0
  4     0
  6     0
  8     0
  9     0
 10     0
 11     0
 12     0
 14     0
 15     0
 16     0
 17     0
 18     0
 19     0
 20     0
 21     0
 22     0
 23     0
 24     0
 70     0
 71     0
 72     0
 73     0
 74     0
 75     0
 76     0
 77     0
 78     0
 80     0
 81     0
```



```
PID COMMAND
  1 /sbin/init
  2 [kthreadd]
  3 [rcu_gp]
  4 [rcu_par_gp]
  6 [kworker/0:0H-kblockd]
  8 [mm_percpu_wq]
  9 [ksoftirqd/0]
 10 [rcu_sched]
 11 [migration/0]
 12 [idle_inject/0]
 14 [cpuhp/0]
 15 [kdevtmpfs]
 16 [netns]
 17 [rcu_tasks_kthre]
 18 [kauditd]
 19 [khungtaskd]
 20 [oom_reaper]
 21 [writeback]
 22 [kcompactd0]
 23 [ksmd]
 24 [khugepaged]
 70 [kintegrityd]
 71 [kblockd]
 72 [blkcg_punt_bio]
 73 [tpm_dev_wq]
 74 [ata_sff]
 75 [md]
 76 [edac-poller]
 77 [devfreq_wq]
 78 [watchdogd]
 80 [kswapd0]
 81 [ecryptfs-kthrea]
```



```
PID COMMAND
  1 systemd
  2 kthreadd
  3 rcu_gp
  4 rcu_par_gp
  6 kworker/0:0H-kblockd
  8 mm_percpu_wq
  9 ksoftirqd/0
 10 rcu_sched
 11 migration/0
 12 idle_inject/0
 14 cpuhp/0
 15 kdevtmpfs
 16 netns
 17 rcu_tasks_kthre
 18 kauditd
 19 khungtaskd
 20 oom_reaper
 21 writeback
 22 kcompactd0
 23 ksmd
 24 khugepaged
 70 kintegrityd
 71 kblockd
 72 blkcg_punt_bio
 73 tpm_dev_wq
 74 ata_sff
 75 md
 76 edac-poller
 77 devfreq_wq
 78 watchdogd
 80 kswapd0
 81 ecryptfs-kthrea
```

Examples of the output if ps -eo is ran to showcase the desired output and information according to the shell script.

Conclusion

The developed shell script could list the currently running processes on the Linux system. The script uses /proc directory as its source of information, parsing through various files in subdirectories to extract and display data such as process ID and other optional information given specific flags. The script also aims to replicate the functionalities of the ps command but with a more specific focus on output columns such as RSS, command name, command line and GID.