

## Operating Systems, Assignment 6

This assignment includes one programming problem. Remember that your work needs to be done individually and that your programs need to be commented and consistently indented. You **must** document your sources and, if you use sources, you must still write at least half your code yourself from scratch.

1. (40 pts) We're going to write a program that simulates the snapshot version of the "top" command in Linux. This command gives overall system information and lists top 10 processes that have the largest memory sizes in descending order. You can type **top** once you log into the EOS Linux machine to see the example results. However, your program does not need to continuously refresh the results like a real top. You just need to print out one snapshot after you type the command. What you need to print does not have to be exactly the same as the real top. Follow the sample output to see the content and format you should print. Besides the overall system information, you should also list the top 10 processes that have the largest memory sizes in descending order. You should find system information for each process using the "/proc" file system. After you get information of all processes, sort them by memory size in descending order and output the top 10. Follow the sample output and make sure that you have printed everything required. I'm giving you a skeleton top.c program to help get you started.

Sample Output Format:

```
[sjiao2@engr-ras-205 HW6]$ ./top
Time: 2023-04-12 18:44:09
Uptime: 3 days, 15:32
Load avg: 1.240000 1.110000 0.870000
Tasks: 891 total 3 running 888 sleeping
Cpu: 2502269us 27111sy 834093ni 183824849id 1514377wa 111528hi 155163si 0st
Mem: 7922188 total 7576808 used 345380 free
Swap: 33550332 total 190208 used 33360124 free
2310785 (cpptools-srv)      S      1      1277859
2350057 (cpptools-srv)      S      1      1277859
2352947 (cpptools-srv)      S      1      1277859
2353783 (cpptools-srv)      S      1      1277859
2310722 (cpptools-srv)      S      1      1277826
1263    (rpc.gssd)          S      1      801126
1104    (polkitd)           S      1      473404
2310435 (cpptools)          S     2310323 389994
2362154 (cpptools)          S     2361943 322301
2310323 (node)              S     2310029 267029
```

Hints:

- Type **man proc** to read the man page of the "/proc" file system. Look over the meaning of each file in this directory and sub-directories. Find where the information you want for this assignment is. You need to deal with each process. To learn how to traverse the /proc directory structure and to read the needed values from the files there, use **man** to review the documentation of the following calls: **opendir**, **readdir**, **fopen**, **fscanf**, **fread**, **fgets**. (You might not need to use all of them. For **readdir**, use **man readdir** and **man 3 readdir** to compare differences).
- The files in /proc/<pid>/ contain the necessary information for individual processes.
- For more information about the whole system, like load average, memory information and so on, you will find some files right in the "/proc" directory to be useful.
- The proc filesystem will not give complete information for some processes. For example, you may get zeros for the memory size of some. You can just ignore these processes (or don't ignore them; they won't make it into the top ten list you're reporting anyway).

- As you are figuring out what values you need out of particular files in `/proc`, feel free to look around and see what the contents of these files look like. You can use the `cd` command to move around the subdirectories in `/proc`, and you can use the `cat` command to echo the contents of a file to your terminal.

When you're done, submit your `top.c` to the assignment named HW6 on Gradescope.

## 2. (No Credit, For Practice Only) Disk Scheduling

Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive has just finished serving a request at cylinder 160, and the previous request was at cylinder 125. The queue of pending requests, in the order they arrived, is:

86, 1470, 913, 1774, 948, 1509, 1022, 2314, 926, 130

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk-scheduling algorithms?

- FCFS
- SSTF
- SCAN
- C-LOOK