Birzeit University

Department of Electrical & Computer Engineering

Second Semester, 2022/2023

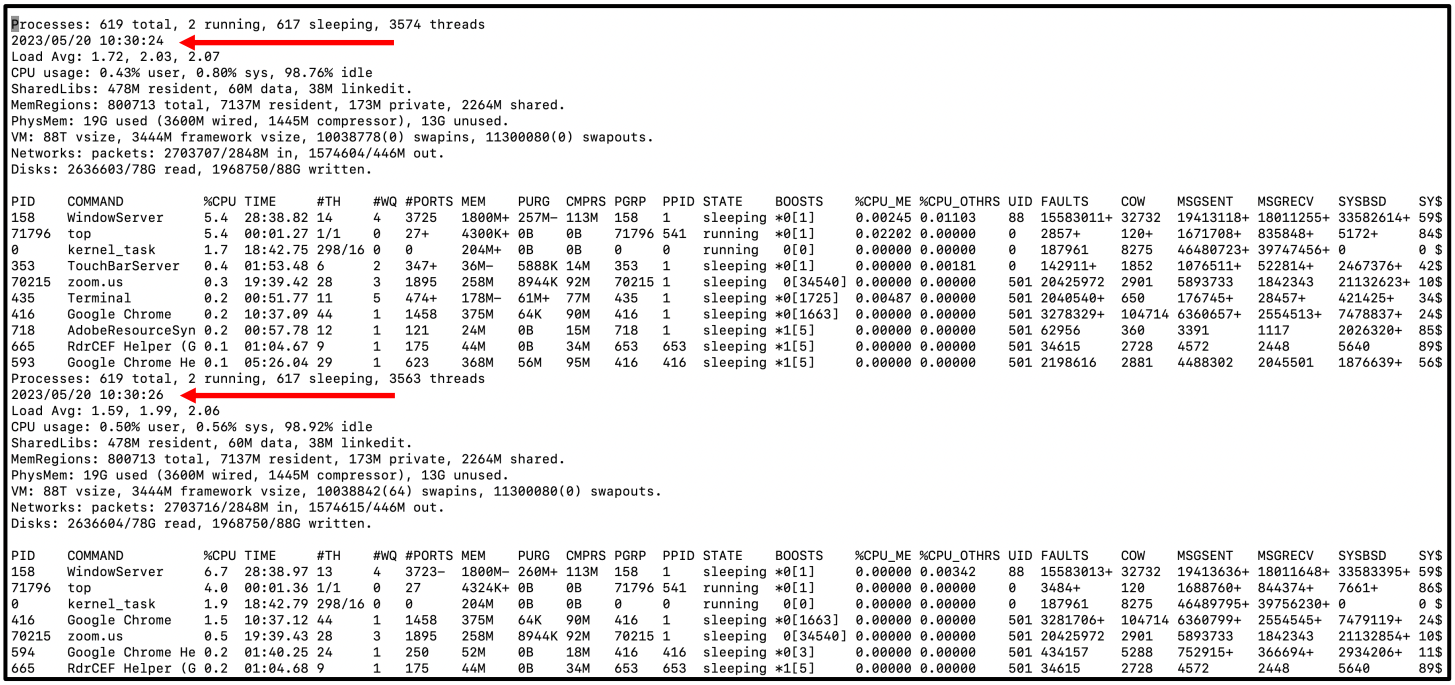
ENCS3130 Linux Laboratory

Shell Scripting Project – Statistics of Running Processes on Linux Machine

You are required to write a shell script that provides statistics about processes running on a Linux computer for this assignment. The top command will be used to get information about the processes that are currently operating. The top command is similar to a CLI version of Windows' task manager. You get a real-time view of the processes and all the information associated with them, such as memory use, CPU usage, and so on.

Input:

Your project's input will be a text file containing the output of the **top** command at various time intervals. Figure 1 shows a screenshot of the first few lines of the text file.

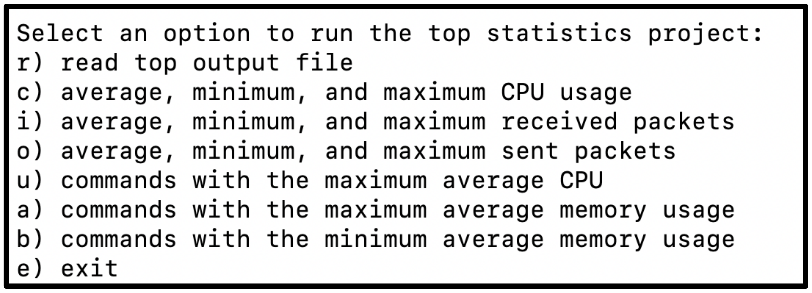


```
Processes: 619 total, 2 running, 617 sleeping, 3574 threads
2023/05/20 10:30:24
Load Avg: 1.72, 2.03, 2.07
CPU usage: 0.43% user, 0.80% sys, 98.76% idle
SharedLibs: 478M resident, 60M data, 38M linkedit.
MemRegions: 800713 total, 7137M resident, 173M private, 2264M shared.
PhysMem: 19G used (3600M wired, 1445M compressor), 13G unused.
VM: 88T vsize, 3444M framework vsize, 10038778(0) swapins, 11300080(0) swapouts.
Networks: packets: 2703707/2848M in, 1574604/446M out.
Disks: 2636603/78G read, 1968750/88G written.

PID    COMMAND          %CPU TIME     #TH    #WQ  #PORTS MEM    PURG  CMPRS PGRP  PPID STATE    BOOSTS    %CPU_ME %CPU_OTHRS UID FAULTS    COW    MSGSENT   MSGRECV   SYSBSD    SY$
158    WindowServer     5.4  28:38.82 14     4    3725   1800M+ 257M- 113M  158   1    sleeping *0[1]     0.00245 0.01103    88  15583011+ 32732  19413118+ 18011255+ 33582614+ 59$
71796  top              5.4  00:01.27 1/1    0    27+    4300K+ 0B    0B    71796 541  running  *0[1]     0.02202 0.00000    0   2857+     120+   1671708+  835848+   5172+     84$
0      kernel_task      1.7  18:42.75 298/16 0    0      204M+  0B    0B    0     0    running   0[0]     0.00000 0.00000    0   187961    8275   46480723+ 39747456+ 0         0 $
353    TouchBarServer   0.4  01:53.48 6      2    347+   36M-   5888K 14M   353   1    sleeping *0[1]     0.00000 0.00181    0   142911+   1852   1076511+  522814+   2467376+  42$
70215  zoom.us          0.3  19:39.42 28     3    1895   258M   8944K 92M   70215 1    sleeping  0[34540] 0.00000 0.00000    501 20425972  2901   5893733   1842343   21132623+ 10$
435    Terminal         0.2  00:51.77 11     5    474+   178M-  61M+  77M   435   1    sleeping *0[1725]  0.00487 0.00000    501 2040540+  650    176745+   28457+    421425+   34$
416    Google Chrome    0.2  10:37.09 44     1    1458   375M   64K   90M   416   1    sleeping *0[1663]  0.00000 0.00000    501 3278329+  104714 6360657+  2554513+  7478837+  24$
718    AdobeResourceSyn 0.2  00:57.78 12     1    121    24M    0B    15M   718   1    sleeping *1[5]     0.00000 0.00000    501 62956     360    3391      1117      2026320+  85$
665    RdrCEF Helper (G 0.1  01:04.67 9      1    175    44M    0B    34M   653   653  sleeping *1[5]     0.00000 0.00000    501 34615     2728   4572      2448      5640      89$
593    Google Chrome He 0.1  05:26.04 29     1    623    368M   56M   95M   416   416  sleeping *1[5]     0.00000 0.00000    501 2198616   2881   4488302   2045501   1876639+  56$
Processes: 619 total, 2 running, 617 sleeping, 3563 threads
2023/05/20 10:30:26
Load Avg: 1.59, 1.99, 2.06
CPU usage: 0.50% user, 0.56% sys, 98.92% idle
SharedLibs: 478M resident, 60M data, 38M linkedit.
MemRegions: 800713 total, 7137M resident, 173M private, 2264M shared.
PhysMem: 19G used (3600M wired, 1445M compressor), 13G unused.
VM: 88T vsize, 3444M framework vsize, 10038842(64) swapins, 11300080(0) swapouts.
Networks: packets: 2703716/2848M in, 1574615/446M out.
Disks: 2636604/78G read, 1968750/88G written.

PID    COMMAND          %CPU TIME     #TH    #WQ  #PORTS MEM    PURG  CMPRS PGRP  PPID STATE    BOOSTS    %CPU_ME %CPU_OTHRS UID FAULTS    COW    MSGSENT   MSGRECV   SYSBSD    SY$
158    WindowServer     6.7  28:38.97 13     4    3723-  1800M- 260M+ 113M  158   1    sleeping *0[1]     0.00000 0.00342    88  15583013+ 32732  19413636+ 18011648+ 33583395+ 59$
71796  top              4.0  00:01.36 1/1    0    27     4324K+ 0B    0B    71796 541  running  *0[1]     0.00000 0.00000    0   3484+     120    1688760+  844374+   7661+     86$
0      kernel_task      1.9  18:42.79 298/16 0    0      204M   0B    0B    0     0    running   0[0]     0.00000 0.00000    0   187961    8275   46489795+ 39756230+ 0         0 $
416    Google Chrome    1.5  10:37.12 44     1    1458   375M   64K   90M   416   1    sleeping *0[1663]  0.00000 0.00000    501 3281706+  104714 6360799+  2554545+  7479119+  24$
70215  zoom.us          0.5  19:39.43 28     3    1895   258M   8944K 92M   70215 1    sleeping  0[34540] 0.00000 0.00000    501 20425972  2901   5893733   1842343   21132854+ 10$
594    Google Chrome He 0.2  01:40.25 24     1    250    52M    0B    18M   416   416  sleeping *0[3]     0.00000 0.00000    501 434157    5288   752915+   366694+   2934206+  11$
665    RdrCEF Helper (G 0.2  01:04.68 9      1    175    44M    0B    34M   653   653  sleeping *1[5]     0.00000 0.00000    501 34615     2728   4572      2448      5640      89$
```

Figure 1: A snapshot of the input text file.

Procedure:

1. The program should print on the screen the main menu and ask the user to select an option

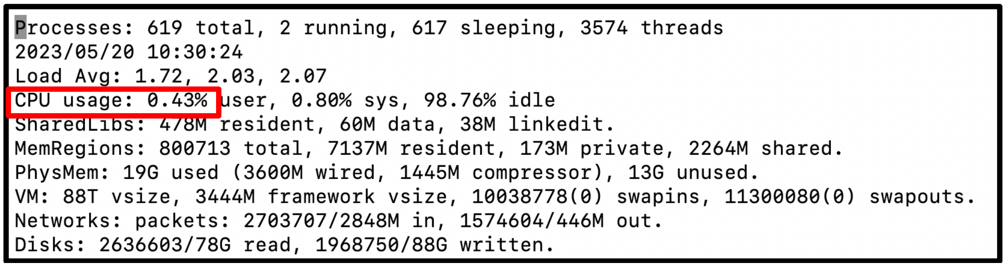


```
Select an option to run the top statistics project:
r) read top output file
c) average, minimum, and maximum CPU usage
i) average, minimum, and maximum received packets
o) average, minimum, and maximum sent packets
u) commands with the maximum average CPU
a) commands with the maximum average memory usage
b) commands with the minimum average memory usage
e) exit
```

Figure 2: the main menu of the project

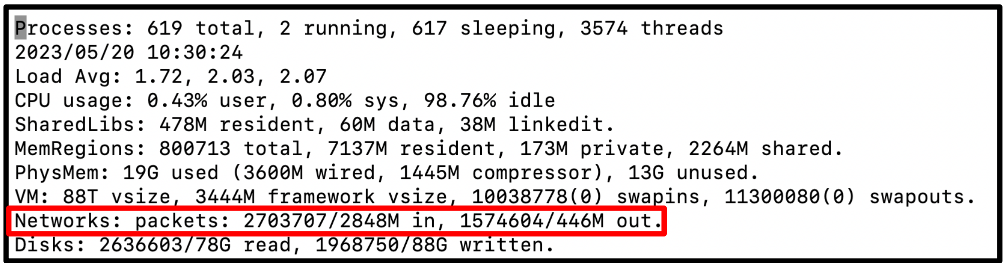
2. If the user enters ‘**r**’:
    a. The program should print on the screen “Please input the name of the file”.
    b. The program should verify that the file exists, otherwise a message should be printed on the screen “file does not exist” and then return to the main menu.

3. If the user enters ‘**c**’: the program should extract the CPU usage from the file for each time instance and then print the average, minimum, and maximum CPU usage on the screen. Also, print the main menu again.



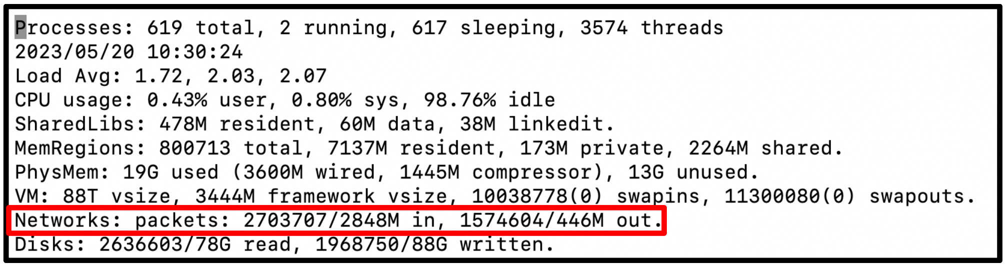
```
Processes: 619 total, 2 running, 617 sleeping, 3574 threads
2023/05/20 10:30:24
Load Avg: 1.72, 2.03, 2.07
CPU usage: 0.43% user, 0.80% sys, 98.76% idle
SharedLibs: 478M resident, 60M data, 38M linkedit.
MemRegions: 800713 total, 7137M resident, 173M private, 2264M shared.
PhysMem: 19G used (3600M wired, 1445M compressor), 13G unused.
VM: 88T vsize, 3444M framework vsize, 10038778(0) swapins, 11300080(0) swapouts.
Networks: packets: 2703707/2848M in, 1574604/446M out.
Disks: 2636603/78G read, 1968750/88G written.
```

4. If the user enters ‘**i**’: the program should extract the number of packets received (in) over the network for each time instance and then print the average, minimum, and maximum received packets. Also, print the main menu again.



```
Processes: 619 total, 2 running, 617 sleeping, 3574 threads
2023/05/20 10:30:24
Load Avg: 1.72, 2.03, 2.07
CPU usage: 0.43% user, 0.80% sys, 98.76% idle
SharedLibs: 478M resident, 60M data, 38M linkedit.
MemRegions: 800713 total, 7137M resident, 173M private, 2264M shared.
PhysMem: 19G used (3600M wired, 1445M compressor), 13G unused.
VM: 88T vsize, 3444M framework vsize, 10038778(0) swapins, 11300080(0) swapouts.
Networks: packets: 2703707/2848M in, 1574604/446M out.
Disks: 2636603/78G read, 1968750/88G written.
```

5. If the user enters ‘**o**’: the program should extract the number of packets sent (out) over the network for each time instance and then print the average, minimum, and maximum sent packets. Also, print the main menu again.



```
Processes: 619 total, 2 running, 617 sleeping, 3574 threads
2023/05/20 10:30:24
Load Avg: 1.72, 2.03, 2.07
CPU usage: 0.43% user, 0.80% sys, 98.76% idle
SharedLibs: 478M resident, 60M data, 38M linkedit.
MemRegions: 800713 total, 7137M resident, 173M private, 2264M shared.
PhysMem: 19G used (3600M wired, 1445M compressor), 13G unused.
VM: 88T vsize, 3444M framework vsize, 10038778(0) swapins, 11300080(0) swapouts.
Networks: packets: 2703707/2848M in, 1574604/446M out.
Disks: 2636603/78G read, 1968750/88G written.
```

6. If the user enters ‘**u**’:
    a. The program should prompt you for an integer number **m**. If the entered value is not an integer, the program should print an error message.
    b. The program should print **m** commands that have the highest maximum average CPU usage on the screen. The program should also print the average CPU usage for each of these commands on the screen. Also, print the main menu again.

7. If the user enters ‘**a**’:
    a. The program should prompt you for an integer number **m**. If the entered value is not an integer, the program should print an error message.
    b. The program should print **m** commands that have the highest average memory usage on the screen. The program should also print the average memory usage for each of these commands on the screen. Also, print the main menu again.

8. If the user enters ‘**b**’:
    a. The program should prompt you for an integer number **m**. If the entered value is not an integer, the program should print an error message.
    b. The program should print **m** commands that have the minimum average memory usage on the screen. The program should also print the average memory usage for each of these commands on the screen. Also, print the main menu again.

9. If the user enters ‘**e**’: the program should print on the screen “Are you sure you want to exist”. If the person inputs “yes”, the program ends. Otherwise, the program should return to the main menu.

Submission:

Please submit the following:

1. Shell script program
2. Report: the report must include the following: a brief introduction, a screen shot of each option with its variations, and add some description of each option execution.

Notes:

- Write the code for the shell script to satisfy the requirements described above and name the script as TopStatistics.sh.
- Make sure your code is clean and well-indented; variables have meaningful names, etc.
- Make sure your script has enough comments inserted to add clarity.
- Work in groups of at most two students
- Deadline: Saturday, 10 June 2023 at 11:59 pm. Please submit your project (code + report) through Ritaj as a reply to this message.
- This project is per group effort: instances of cheating will result in you failing the lab.