Experiment No.1
Analyzing Linux based computer systems using a. top, b. ps, c. kill, d. cat /proc/cpuinfo, e.vmstat
Date of Performance:11/01/2024
Date of Submission:18/04/2024

Aim: To analyze Linux based computer systems using following commands: **a.** top , **b.**ps , **c.** kill, **d.** cat /proc/cpuinfo **e.**vmstat

Objective: To understand linux commands

Theory:

The following commands are used for:

a. top: The 'top' command is used to display real-time system statistics, such as CPU usage, memory usage, running processes, etc. It displays the most CPU-intensive tasks at the top of the list, which helps in identifying and troubleshooting performance issues. The 'top' command can also be used to kill processes that are causing problems.

b. ps: The 'ps' command displays a list of currently running processes on the system. It can be used to find out the process ID (PID), the amount of memory and CPU usage, and the user who initiated the process. The 'ps' command can also be used to terminate a process using the 'kill' command.

c. kill: The 'kill' command is used to terminate a running process. It sends a signal to the process to stop execution. The signal can be specified using different options, such as '-9' to forcefully terminate the process or '-15' to ask the process to terminate gracefully.

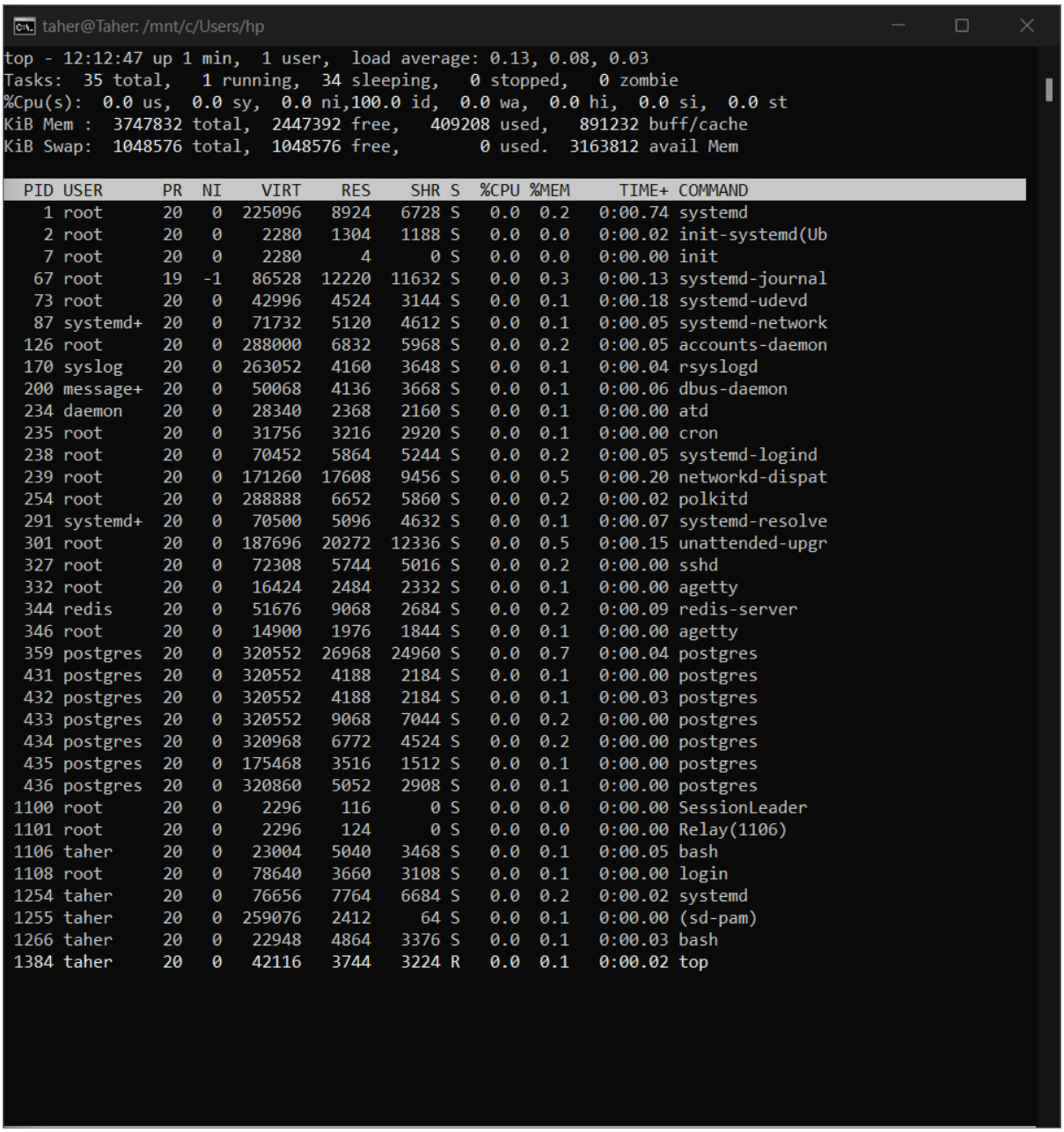
d. cat /proc/cpuinfo: The 'cat /proc/cpuinfo' command displays information about the CPU installed on the system, such as the model name, clock speed, cache size, etc. This information can be useful in determining the capabilities of the system and whether it meets the requirements for running certain applications.

e. vmstat: The 'vmstat' command is used to display information about the virtual memory system of the system. It provides information on the amount of free memory, the number of processes waiting for I/O, the amount of swap space used, etc. This information can be used to diagnose performance issues related to memory and disk usage.

These commands can be used to gather information about the CPU, memory usage, running processes, and system performance. With these commands, system administrators can troubleshoot performance issues and optimize system resources for better efficiency. Familiarity with these commands is essential for anyone who manages Linux systems.

Output:

Top:


taher@Taher: /mnt/c/Users/hp
top - 12:12:47 up 1 min, 1 user, load average: 0.13, 0.08, 0.03
Tasks: 35 total, 1 running, 34 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 3747832 total, 2447392 free, 409208 used, 891232 buff/cache
KiB Swap: 1048576 total, 1048576 free, 0 used. 3163812 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1 root 20 0 225096 8924 6728 S 0.0 0.2 0:00.74 systemd
2 root 20 0 2280 1304 1188 S 0.0 0.0 0:00.02 init-systemd(Ub
7 root 20 0 2280 4 0 S 0.0 0.0 0:00.00 init
67 root 19 -1 86528 12220 11632 S 0.0 0.3 0:00.13 systemd-journal
73 root 20 0 42996 4524 3144 S 0.0 0.1 0:00.18 systemd-udevd
87 systemd+ 20 0 71732 5120 4612 S 0.0 0.1 0:00.05 systemd-network
126 root 20 0 288000 6832 5968 S 0.0 0.2 0:00.05 accounts-daemon
170 syslog 20 0 263052 4160 3648 S 0.0 0.1 0:00.04 rsyslogd
200 message+ 20 0 50068 4136 3668 S 0.0 0.1 0:00.06 dbus-daemon
234 daemon 20 0 28340 2368 2160 S 0.0 0.1 0:00.00 atd
235 root 20 0 31756 3216 2920 S 0.0 0.1 0:00.00 cron
238 root 20 0 70452 5864 5244 S 0.0 0.2 0:00.05 systemd-logind
239 root 20 0 171260 17608 9456 S 0.0 0.5 0:00.20 networkd-dispat
254 root 20 0 288888 6652 5860 S 0.0 0.2 0:00.02 polkitd
291 systemd+ 20 0 70500 5096 4632 S 0.0 0.1 0:00.07 systemd-resolve
301 root 20 0 187696 20272 12336 S 0.0 0.5 0:00.15 unattended-upgr
327 root 20 0 72308 5744 5016 S 0.0 0.2 0:00.00 sshd
332 root 20 0 16424 2484 2332 S 0.0 0.1 0:00.00 agetty
344 redis 20 0 51676 9068 2684 S 0.0 0.2 0:00.09 redis-server
346 root 20 0 14900 1976 1844 S 0.0 0.1 0:00.00 agetty
359 postgres 20 0 320552 26968 24960 S 0.0 0.7 0:00.04 postgres
431 postgres 20 0 320552 4188 2184 S 0.0 0.1 0:00.00 postgres
432 postgres 20 0 320552 4188 2184 S 0.0 0.1 0:00.03 postgres
433 postgres 20 0 320552 9068 7044 S 0.0 0.2 0:00.00 postgres
434 postgres 20 0 320968 6772 4524 S 0.0 0.2 0:00.00 postgres
435 postgres 20 0 175468 3516 1512 S 0.0 0.1 0:00.00 postgres
436 postgres 20 0 320860 5052 2908 S 0.0 0.1 0:00.00 postgres
1100 root 20 0 2296 116 0 S 0.0 0.0 0:00.00 SessionLeader
1101 root 20 0 2296 124 0 S 0.0 0.0 0:00.00 Relay(1106)
1106 taher 20 0 23004 5040 3468 S 0.0 0.1 0:00.05 bash
1108 root 20 0 78640 3660 3108 S 0.0 0.1 0:00.00 login
1254 taher 20 0 76656 7764 6684 S 0.0 0.2 0:00.02 systemd
1255 taher 20 0 259076 2412 64 S 0.0 0.1 0:00.00 (sd-pam)
1266 taher 20 0 22948 4864 3376 S 0.0 0.1 0:00.03 bash
1384 taher 20 0 42116 3744 3224 R 0.0 0.1 0:00.02 top

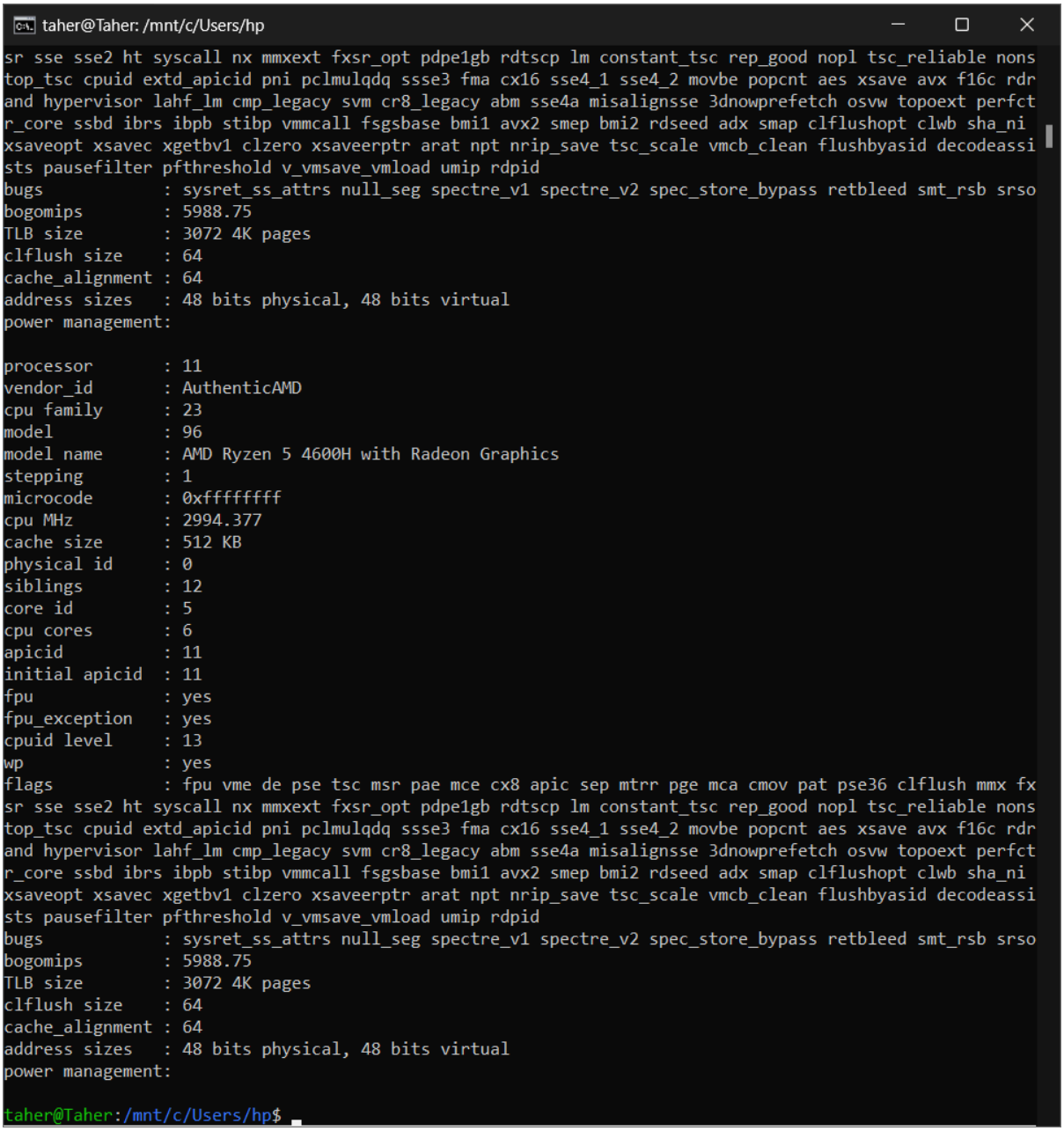
PS:

```
taher@Taher:/mnt/c/Users/hp$ ps
  PID TTY          TIME CMD
 1106 pts/0    00:00:00 bash
 1391 pts/0    00:00:00 ps
```

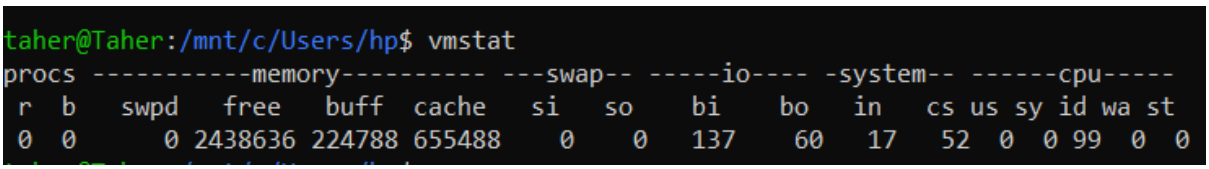
Kill:

```
taher@Taher:/mnt/c/Users/hp$ sudo kill 359
[sudo] password for taher:
taher@Taher:/mnt/c/Users/hp$
```

Cat /proc/cpuinfo:



vmstat:



Conclusion: In conclusion, the examination of Linux-based computer systems through commands such as 'top', 'ps', 'kill', 'cat /proc/cpuinfo', and 'vmstat' offers a comprehensive insight into system performance and resource utilization. These commands empower users to monitor processes, manage system resources efficiently, and troubleshoot issues effectively. By leveraging these tools, administrators can ensure optimal system functionality, enhance performance, and maintain stability in Linux environments.