

# Operating System

## Lab-2



### System Monitoring in Linux



### Commands

1. top: The top command in Linux is a real-time system monitoring tool that provides an interactive and dynamic view of the system resource utilization. It provides information about various system metrics, including CPU usage, memory usage, load averages, running processes and more.

Usage: `top -n 5 d -2`

This command runs top command for 5 iterations with a 2 sec delay between updates.

```
Activities Terminal Aug 15 18:14
vboxuser@blockexplorer: ~
Tasks: 176 total, 1 running, 175 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.7 us, 4.1 sy, 0.0 ni, 90.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1963.5 total, 513.9 free, 669.9 used, 779.7 buff/cache
MiB Swap: 2680.0 total, 2680.0 free, 0.0 used. 1116.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1448 vboxuser  20   0 3516596 333124 135820 S   5.4  16.6  0:23.83 gnome-+
 1938 vboxuser  20   0 570644 52612 40064 S   2.5   2.6  0:04.34 gnome-+
 2115 vboxuser  20   0 21728 4072 3380 R   1.5   0.2  0:00.12 top
 579 systemd+ 20   0 14828 6256 5460 S   0.5   0.3  0:02.45 system+
1592 vboxuser  20   0 323652 12108 7156 S   0.5   0.6  0:01.39 ibus-d+
   1 root      20   0 166760 11944 8312 S   0.0   0.6  0:02.10 systemd
   2 root      20   0      0      0      0 S   0.0   0.0  0:00.00 kthrea+
   3 root      0 -20      0      0      0 I   0.0   0.0  0:00.00 rcu_gp
   4 root      0 -20      0      0      0 I   0.0   0.0  0:00.00 rcu_pa+
   5 root      0 -20      0      0      0 I   0.0   0.0  0:00.00 slub_f+
   6 root      0 -20      0      0      0 I   0.0   0.0  0:00.00 netns
   8 root      0 -20      0      0      0 I   0.0   0.0  0:00.00 kworke+
   9 root      20   0      0      0      0 I   0.0   0.0  0:01.24 kworke+
  10 root      0 -20      0      0      0 I   0.0   0.0  0:00.00 mm_per+
  11 root      20   0      0      0      0 I   0.0   0.0  0:00.00 rcu_ta+
  12 root      20   0      0      0      0 I   0.0   0.0  0:00.00 rcu_ta+
  13 root      20   0      0      0      0 I   0.0   0.0  0:00.00 rcu_ta+
  14 root      20   0      0      0      0 S   0.0   0.0  0:00.20 ksofti+
  15 root      20   0      0      0      0 I   0.0   0.0  0:00.69 rcu_pr+
  16 root      rt   0      0      0      0 S   0.0   0.0  0:00.01 migrat+
  17 root     -51   0      0      0      0 S   0.0   0.0  0:00.00 idle_i+
  18 root      20   0      0      0      0 I   0.0   0.0  0:01.08 kworke+

vboxuser@blockexplorer: ~$
```

2. ps: The ps (process status) command is used to display all the running processes in the Linux system.

In the output of ps the fields are:

1. PID: Unique Process id
2. TTY: terminal type of the user logged into.
3. Time: amount of CPU time in minutes and seconds that the process has been running.
4. CMD: name of the command that launched the process.

Usage: ps -e

```
Activities Terminal Aug 15 18:17 vboxuser@blockexplorer: ~  
vboxuser@blockexplorer:~$ ps  
  PID TTY          TIME CMD  
 1960 pts/0    00:00:00 bash  
 2118 pts/0    00:00:00 ps  
vboxuser@blockexplorer:~$ ps -A  
  PID TTY          TIME CMD  
    1 ?           00:00:02 systemd  
    2 ?           00:00:00 kthreadd  
    3 ?           00:00:00 rcu_gp  
    4 ?           00:00:00 rcu_par_gp  
    5 ?           00:00:00 slub_flushwq  
    6 ?           00:00:00 netns  
    8 ?           00:00:00 kworker/0:0H-events_highpri  
    9 ?           00:00:01 kworker/u2:0-events_power_efficient  
   10 ?           00:00:00 mm_percpu_wq  
   11 ?           00:00:00 rcu_tasks_kthread  
   12 ?           00:00:00 rcu_tasks_rude_kthread  
   13 ?           00:00:00 rcu_tasks_trace_kthread  
   14 ?           00:00:00 ksoftirqd/0  
   15 ?           00:00:00 rcu_preempt  
   16 ?           00:00:00 migration/0  
   17 ?           00:00:00 idle_inject/0  
   18 ?           00:00:01 kworker/0:1-events  
   19 ?           00:00:00 cpuhp/0  
   20 ?           00:00:00 kdevtmpfs  
   21 ?           00:00:00 inet_frag_wq  
   22 ?           00:00:00 kauditd  
   23 ?           00:00:00 khungtaskd  
   25 ?           00:00:00 oom_reaper
```

```
Activities Terminal Aug 15 18:20 vboxuser@blockexplorer: ~  
2122 pts/0    00:00:00 ps  
vboxuser@blockexplorer:~$ ps -ef  
  UID      PID  PPID  C  STIME TTY          TIME CMD  
root         1      0  0  18:02 ?          00:00:02 /sbin/init splash  
root         2      0  0  18:02 ?          00:00:00 [kthreadd]  
root         3      2  0  18:02 ?          00:00:00 [rcu_gp]  
root         4      2  0  18:02 ?          00:00:00 [rcu_par_gp]  
root         5      2  0  18:02 ?          00:00:00 [slub_flushwq]  
root         6      2  0  18:02 ?          00:00:00 [netns]  
root         8      2  0  18:02 ?          00:00:00 [kworker/0:0H-events_highpri  
root         9      2  0  18:02 ?          00:00:01 [kworker/u2:0-events_power_  
root        10      2  0  18:02 ?          00:00:00 [mm_percpu_wq]  
root        11      2  0  18:02 ?          00:00:00 [rcu_tasks_kthread]  
root        12      2  0  18:02 ?          00:00:00 [rcu_tasks_rude_kthread]  
root        13      2  0  18:02 ?          00:00:00 [rcu_tasks_trace_kthread]  
root        14      2  0  18:02 ?          00:00:00 [ksoftirqd/0]  
root        15      2  0  18:02 ?          00:00:00 [rcu_preempt]  
root        16      2  0  18:02 ?          00:00:00 [migration/0]  
root        17      2  0  18:02 ?          00:00:00 [idle_inject/0]  
root        18      2  0  18:02 ?          00:00:01 [kworker/0:1-events]  
root        19      2  0  18:02 ?          00:00:00 [cpuhp/0]  
root        20      2  0  18:02 ?          00:00:00 [kdevtmpfs]  
root        21      2  0  18:02 ?          00:00:00 [inet_frag_wq]  
root        22      2  0  18:02 ?          00:00:00 [kauditd]  
root        23      2  0  18:02 ?          00:00:00 [khungtaskd]  
root        25      2  0  18:02 ?          00:00:00 [oom_reaper]  
root        26      2  0  18:02 ?          00:00:00 [writeback]  
root        28      2  0  18:02 ?          00:00:00 [kcompactd0]  
root        29      2  0  18:02 ?          00:00:00 [ksmd]
```

3. **iostat:** The `iostat` command in Linux is used to monitor and display the input/output statistics of block devices and partitions. It provides valuable information about the performance and utilization of various storage devices including hard drives, solid-state drives and other storage devices. This can help administrators and users to identify potential bottlenecks, monitor disk activity and optimize system performance. The `iostat` command provides information about various aspects of I/O performance including:

1. **Disk Utilization:** Displays the percentage of time a device's disk is active and serving I/O requests.
2. **Disk I/O operation:** Number of read and write operations being performed on the disk per second.
3. **Average Wait Time:** Provides information about the average time an I/O request spends in queue waiting to be served.
4. **Transfer Rates:** The amount of data being read or written to the disk per second.
5. **CPU Utilization:** How much of CPU's process power is being used to handle I/O operation.

```
Processing triggers for Man-db (2.10.2-1) ...
kartik@ubuntu-Kartik:~$ iostat
Linux 5.15.0-56-generic (ubuntu-Kartik)      15/08/23      _x86_64_      (12 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.13    0.04   0.18   1.09    0.00   98.55

Device            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
loop0              0.03         0.24         0.00         0.00         351         0         0
loop1              0.01         0.01         0.00         0.00         17         0         0
loop10             0.04         0.77         0.00         0.00        1101         0         0
loop11             0.04         0.74         0.00         0.00        1069         0         0
loop12             0.79         9.99         0.00         0.00       14377         0         0
loop13             0.41         4.35         0.00         0.00        6260         0         0
loop14             0.03         0.24         0.00         0.00         351         0         0
loop15             0.03         0.25         0.00         0.00         361         0         0
loop16             0.03         0.24         0.00         0.00         348         0         0
loop17             0.51        17.71         0.00         0.00       25489         0         0
loop18             0.01         0.04         0.00         0.00          54         0         0
loop19             0.02         0.18         0.00         0.00         260         0         0
loop2              0.03         0.25         0.00         0.00         361         0         0
loop20             0.01         0.01         0.00         0.00          14         0         0
loop3              0.04         0.74         0.00         0.00        1072         0         0
loop4              0.25         3.13         0.00         0.00        4500         0         0
loop5              0.04         0.75         0.00         0.00        1077         0         0
loop6              0.04         0.76         0.00         0.00        1099         0         0
loop7              0.04         0.75         0.00         0.00        1080         0         0
loop8              0.04         0.77         0.00         0.00        1110         0         0
loop9              0.04         0.75         0.00         0.00        1083         0         0
sda               15.48        516.31        39.21         0.00      743074      56433         0
sr0               0.05         0.11         0.00         0.00         154         0         0

kartik@ubuntu-Kartik:~$
```

4. strace: The strace command in Linux is used to trace and analyze the system calls and the signals made by a process during its execution. It provides a detailed view of the interactions between a user and the operating system, showing the sequence of system calls, their arguments and their return values. This can be useful for debugging, profiling and understanding how a program interacts with the underlying system.

Usage: *strace -o output.txt -f -e trace=write,open ls -l*

-o output.txt : specifies that the output should be saved to a file named output.txt.

-f: follows child process

-e traces=write,open: specifies which system calls to trace. In this case it is tracing the write and open system calls.

ls -l: denotes the command you want to trace.

```
strace -f -e trace=write,open ls -l
vboxuser@blockexplorer:/$ strace -f -e trace=write,open ls -l
write(1, "total 2744400\n", 14total 2744400
) = 14
write(1, "lrwxrwxrwx  1 root root      "..., 64lrwxrwxrwx  1 root root
      7 Aug  1 10:55 bin -> usr/bin
) = 64
write(1, "drwxr-xr-x  4 root root      4"..., 54drwxr-xr-x  4 root root
      4096 Aug  1 11:43 boot
) = 54
write(1, "drwxrwxr-x  2 root root      4"..., 55drwxrwxr-x  2 root root
      4096 Aug  1 11:19 cdrom
) = 55
write(1, "drwxr-xr-x 19 root root      4"..., 53drwxr-xr-x 19 root root
      4260 Aug 15 18:03 dev
) = 53
write(1, "drwxr-xr-x 130 root root     12"..., 53drwxr-xr-x 130 root root
      12288 Aug  1 11:49 etc
) = 53
write(1, "drwxr-xr-x  3 root root      4"..., 54drwxr-xr-x  3 root root
      4096 Aug  1 11:20 home
) = 54
write(1, "lrwxrwxrwx  1 root root      "..., 64lrwxrwxrwx  1 root root
      7 Aug  1 10:55 lib -> usr/lib
) = 64
write(1, "lrwxrwxrwx  1 root root      "..., 68lrwxrwxrwx  1 root root
      9 Aug  1 10:55 lib32 -> usr/lib32
) = 68
```

5. Isof: The Isof command (short for list open files) command in Linux is used to display information about the files that are currently open by various processes on the system. This command provides information to which files or new sockets are being used by the running processes, along with the details about the processes themselves. It helps in administrating, troubleshooting and identifying resource usage.

Usage: Isof -i TCP:80 : This command lists all the processes with open TCP sockets on port80, which is commonly used for HTTP traffic. This helps to see which processes are listening on or connecting to that port.

```
dbus-daem 1321 vboxuser 31u      unix 0x0000000000000000      0t0      23820
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 32u      unix 0x0000000000000000      0t0      24069
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 33u      unix 0x0000000000000000      0t0      23833
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 34u      unix 0x0000000000000000      0t0      23954
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 35u      unix 0x0000000000000000      0t0      24009
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 36u      unix 0x0000000000000000      0t0      24037
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 38u      unix 0x0000000000000000      0t0      25839
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 39u      unix 0x0000000000000000      0t0      24526
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 40u      unix 0x0000000000000000      0t0      24084
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 41u      unix 0x0000000000000000      0t0      24101
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 42u      unix 0x0000000000000000      0t0      24172
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 43u      unix 0x0000000000000000      0t0      24130
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 44u      unix 0x0000000000000000      0t0      24219
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 45u      unix 0x0000000000000000      0t0      24236
/run/user/1000/bus type=STREAM
dbus-daem 1321 vboxuser 46u      unix 0x0000000000000000      0t0      24237
```



6. **lsblk**: The **lsblk** command is used to display information about block devices, which includes storage devices such as hard drives, solid state drives, and other devices that use block I/O subsystem.

Usage: **lsblk**

```
vboxuser@blockexplorer:/$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0    4K  1 loop /snap/bare/5
loop1        7:1      0  63.3M  1 loop /snap/core20/1822
loop2        7:2      0  63.4M  1 loop /snap/core20/1974
loop3        7:3      0  73.9M  1 loop /snap/core22/817
loop4        7:4      0  73.9M  1 loop /snap/core22/858
loop5        7:5      0 240.6M  1 loop /snap/firefox/2356
loop6        7:6      0 237.2M  1 loop /snap/firefox/2987
loop7        7:7      0 346.3M  1 loop /snap/gnome-3-38-2004/119
loop8        7:8      0 349.7M  1 loop /snap/gnome-3-38-2004/143
loop9        7:9      0 485.5M  1 loop /snap/gnome-42-2204/120
loop10       7:10     0 485.5M  1 loop /snap/gnome-42-2204/126
loop11       7:11     0  91.7M  1 loop /snap/gtk-common-themes/1535
loop12       7:12     0  45.9M  1 loop /snap/snap-store/638
loop13       7:13     0  12.3M  1 loop /snap/snap-store/959
loop14       7:14     0  49.8M  1 loop /snap/snapd/18357
loop15       7:15     0  53.3M  1 loop /snap/snapd/19457
loop16       7:16     0   304K  1 loop /snap/snapd-desktop-integration/49
loop17       7:17     0   452K  1 loop /snap/snapd-desktop-integration/83
sda          8:0      0   25G   0 disk
├─sda1       8:1      0    1M   0 part
├─sda2       8:2      0   513M  0 part /boot/efi
└─sda3       8:3      0  24.5G  0 part /var/snap/firefox/common/host-hunspell
```

## ➤ Proc File System

A proc file system is a temporary virtual file system which is established when the system boots up and is dissolved when the system shuts down. The proc file system serves as a command and information hub for the Kernel and



provides essential information about the processes that are currently active. A channel of communication between Kernel and the user space is also offered via the proc file system.

Syntax: `ls -l /proc`

This command will list down all the files and directories under the '/proc' directory with detailed information like permissions, ownership, size and time of modification useful for understanding the current status of our system and diagnosing the problems.