



ALBUKHARY INTERNATIONAL UNIVERSITY

## SCHOOL OF COMPUTING & INFORMATICS

---

Course Title : **OPERATING SYSTEMS**  
Course Code : **CCC 2123**

---

### LAB EXERCISE (WEEK 9)

---

#### Instructions:

Using the Command Line Interface (CLI) for Linux that we have learned and used so far, please execute the following tasks:

#### **Task 1: System Information:**

- Display the system's hostname.

hostname

- Retrieve and present information about the kernel version.

uname -a

```
Other Linux 5.x kernel - VMware Workstation 17 Player (Non-commercial use only)
Player
(haryani@kali)-[~]
$ hostname
kali
(haryani@kali)-[~]
$ uname -a
Linux kali 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) x86_64 GNU/Linux
```

✓ When you run `uname -a`, it outputs a single line with the following information:

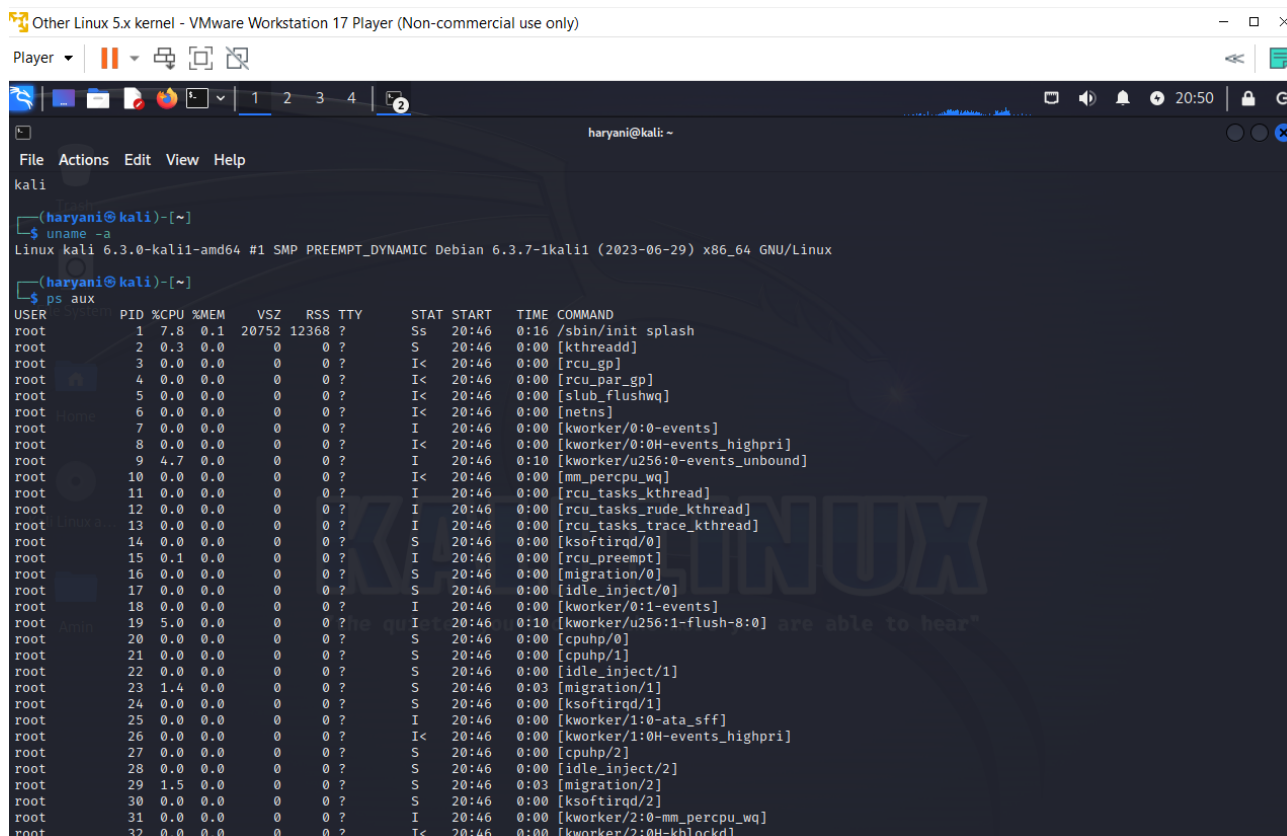
- System Name
- Kernel Release
- Kernel Version
- Machine Architecture
- Operating System Type
- Hostname

#The `uname -a` command is useful for quickly obtaining detailed information about the operating system and kernel version running on a system.

## Task 2: Process Management:

- List all currently running processes.

ps aux



```
haryani@kali: ~  
$ ps aux  
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root           1  7.8  0.1 20752 12368 ?        Ss   20:46   0:16 /sbin/init splash  
root           2  0.3  0.0      0     0 ?        S    20:46   0:00 [kthreadd]  
root           3  0.0  0.0      0     0 ?        I<   20:46   0:00 [rcu_gp]  
root           4  0.0  0.0      0     0 ?        I<   20:46   0:00 [rcu_par_gp]  
root           5  0.0  0.0      0     0 ?        I<   20:46   0:00 [slub_flushwq]  
root           6  0.0  0.0      0     0 ?        I<   20:46   0:00 [netns]  
root           7  0.0  0.0      0     0 ?        I    20:46   0:00 [kworker/0:0-events]  
root           8  0.0  0.0      0     0 ?        I<   20:46   0:00 [kworker/0:0H-events_highpri]  
root           9  4.7  0.0      0     0 ?        I    20:46   0:10 [kworker/u256:0-events_unbound]  
root          10  0.0  0.0      0     0 ?        I<   20:46   0:00 [mm_percpu_wq]  
root          11  0.0  0.0      0     0 ?        I    20:46   0:00 [rcu_tasks_kthread]  
root          12  0.0  0.0      0     0 ?        I    20:46   0:00 [rcu_tasks_rude_kthread]  
root          13  0.0  0.0      0     0 ?        I    20:46   0:00 [rcu_tasks_trace_kthread]  
root          14  0.0  0.0      0     0 ?        S    20:46   0:00 [ksoftirqd/0]  
root          15  0.1  0.0      0     0 ?        I    20:46   0:00 [rcu_preempt]  
root          16  0.0  0.0      0     0 ?        S    20:46   0:00 [migration/0]  
root          17  0.0  0.0      0     0 ?        S    20:46   0:00 [idle_inject/0]  
root          18  0.0  0.0      0     0 ?        I    20:46   0:00 [kworker/0:1-events]  
root          19  5.0  0.0      0     0 ?        I    20:46   0:10 [kworker/u256:1-flush-8:0]  
root          20  0.0  0.0      0     0 ?        S    20:46   0:00 [cpuhp/0]  
root          21  0.0  0.0      0     0 ?        S    20:46   0:00 [cpuhp/1]  
root          22  0.0  0.0      0     0 ?        S    20:46   0:00 [idle_inject/1]  
root          23  1.4  0.0      0     0 ?        S    20:46   0:03 [migration/1]  
root          24  0.0  0.0      0     0 ?        S    20:46   0:00 [ksoftirqd/1]  
root          25  0.0  0.0      0     0 ?        I    20:46   0:00 [kworker/1:0-ata_sff]  
root          26  0.0  0.0      0     0 ?        I<   20:46   0:00 [kworker/1:0H-events_highpri]  
root          27  0.0  0.0      0     0 ?        S    20:46   0:00 [cpuhp/2]  
root          28  0.0  0.0      0     0 ?        S    20:46   0:00 [idle_inject/2]  
root          29  1.5  0.0      0     0 ?        S    20:46   0:03 [migration/2]  
root          30  0.0  0.0      0     0 ?        S    20:46   0:00 [ksoftirqd/2]  
root          31  0.0  0.0      0     0 ?        I    20:46   0:00 [kworker/2:0-mm_percpu_wq]  
root          32  0.0  0.0      0     0 ?        I<   20:46   0:00 [kworker/2:0H-kblockd]
```

- ps: Stands for "process status." It is a command used to provide information about the currently running processes on a system.
- aux:
  - ✓ a: Lists the processes of all users. Without this option, ps shows only the processes associated with the current user.
  - ✓ u: Displays additional information, including the user who owns the process (USER column), the CPU and memory usage (%CPU and %MEM columns), and the start time of the process (START column).
  - ✓ x: Adds processes not attached to a terminal.

So, when you run ps aux, you get a comprehensive list of all processes running on the system, along with detailed information such as the user who owns the process, the process ID (PID), the percentage of CPU and memory usage, the start time of the process, and more.

#This command is useful for monitoring system activity and identifying resource-intensive processes.

- Find and display information about the process associated with the terminal.

echo \$\$

```
(haryani@kali)-[~]
$ echo $$
2056
(haryani@kali)-[~]
$
```

- ✓ The echo \$\$ command in a Unix-like shell, such as Bash, prints the Process ID (PID) of the currently running shell or script.
- ✓ echo: A command used to print the value of a variable or expression.
- ✓ \$\$: In the context of the echo command, \$\$ represents a special variable that holds the PID of the current process.
- ✓ So, when you run echo \$\$, the shell replaces \$\$ with the actual PID of the running shell or script and then prints that PID to the terminal.
- ✓ For example, if the output is 2056, it means that the PID of the current shell or script is 2056.

# This information can be useful in various scenarios, such as tracking or managing processes, especially within scripts where you may need to reference the PID programmatically.

- Terminate a specific process using the appropriate command.

kill 2056

- Check whether the process has been terminated.

ps aux | grep 2056

```
(haryani@kali)-[~]
$ kill 2056

(haryani@kali)-[~]
$ ps aux | grep 2056
haryani  2056  0.1  0.0 10668 6364 pts/1    Ss   20:48   0:01 /usr/bin/zsh
haryani  9225  0.0  0.0  6868 1920 pts/1    S+   21:02   0:00 grep --color=auto 2056
```

### Task 3: System Resource Usage:

- Utilize a command to show the current system load.

uptime

```
(haryani@kali)-[~]  
$ uptime  
21:06:20 up 19 min, 1 user, load average: 0.22, 0.20, 0.19
```

- ✓ When you run the uptime command, it typically displays a single line of output with the following information:

- Current Time
- System Uptime
- Number of Users
- Load Averages

- Identify the top processes consuming CPU and memory resources.

top

```
(haryani@kali)-[~]  
$ top  
top - 21:07:09 up 20 min, 1 user, load average: 0.25, 0.20, 0.19  
Tasks: 204 total, 1 running, 203 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.3 us, 1.1 sy, 0.0 ni, 98.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 7904.7 total, 6663.2 free, 1016.2 used, 481.6 buff/cache  
MiB Swap: 975.0 total, 975.0 free, 0.0 used, 6888.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
909	root	20	0	443604	122904	69344	S	2.0	1.5	0:20.42	Xorg
2045	haryani	20	0	452036	109536	88336	S	1.0	1.4	0:06.15	qterminal
591	root	20	0	241656	12796	7424	S	0.7	0.2	0:04.10	vmtoolsd
1347	haryani	20	0	432284	28068	20768	S	0.7	0.3	0:05.80	panel-15-genmon
6819	root	20	0	0	0	0	I	0.7	0.0	0:01.09	kworker/1:2-pm
15	root	20	0	0	0	0	I	0.3	0.0	0:01.05	rcu_preempt
40	root	20	0	0	0	0	I	0.3	0.0	0:15.74	kworker/u256:2-ext4-rsv-conversion
44	root	20	0	0	0	0	I	0.3	0.0	0:00.44	kworker/2:1-events
1285	haryani	20	0	1315040	103480	77860	S	0.3	1.3	0:06.97	xfwm4
1345	haryani	20	0	284148	34280	19328	S	0.3	0.4	0:06.15	panel-13-cpugra
1482	haryani	20	0	363544	41520	29756	S	0.3	0.5	0:03.83	vmtoolsd
11260	haryani	20	0	12156	5120	2944	R	0.3	0.1	0:00.05	top
1	root	20	0	20752	12368	9296	S	0.0	0.2	0:16.86	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.68	kthreadd
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_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
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	kworker/0:0H-events_highpri
9	root	20	0	0	0	0	I	0.0	0.0	0:10.34	kworker/u256:0-writeback
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
14	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksftirad/0

- ✓ When you run the top command in the terminal, it displays a real-time, dynamic view of the system's processes and their resource utilization.
- ✓ The reason the list changes from time to time is because top continuously updates the information to provide a live snapshot of the system's activity.
- ✓ You can interact with top by using various commands (e.g., pressing 'k' to kill a process or 'q' to quit).
- ✓ This allows users to take actions based on the real-time information provided.