

1. Use the ps cmd to list down all the processes ,pipe it to more or less for paging

a. ps -e

```
[jaysimman@vmware ~]$ ps -e
```

PID	TTY	TIME	CMD
1	?	00:00:13	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	pool_workqueue_
4	?	00:00:00	kworker/R-rcu_g
5	?	00:00:00	kworker/R-rcu_p
6	?	00:00:00	kworker/R-slab_
7	?	00:00:00	kworker/R-netns
9	?	00:00:00	kworker/0:0H-xfst-log/dm-0
11	?	00:00:00	kworker/R-mm_pe
13	?	00:00:00	rcu_tasks_kthre
14	?	00:00:00	rcu_tasks_rude_
15	?	00:00:00	rcu_tasks_trace
16	?	00:00:02	ksoftirqd/0
17	?	00:00:01	rcu_preempt
18	?	00:00:00	migration/0

b. ps -ef

```
[jaysimman@vmware ~]$ ps -ef | head -5
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	11:29	?	00:00:13	/usr/lib/systemd/systemd rhgb --switched-root --system --deserialize 31
root	2	0	0	11:29	?	00:00:00	[kthreadd]
root	3	2	0	11:29	?	00:00:00	[pool_workqueue_]
root	4	2	0	11:29	?	00:00:00	[kworker/R-rcu_g]

```
[jaysimman@vmware ~]$
```

c. ps -f

```
[jaysimman@vmware ~]$ ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
jaysimm+	3894	3893	0	15:07	pts/0	00:00:00	-bash
jaysimm+	4032	3894	0	15:26	pts/0	00:00:00	cat
jaysimm+	4100	4099	0	15:34	pts/0	00:00:00	bash
jaysimm+	4129	4100	0	15:36	pts/0	00:00:00	ps -f

d. ps -aux

```
[jaysimman@vmware ~]$ ps aux | tail -5
```

USER	PPID	PID	%CPU	%MEM	VSZ	SSZ	TTY	TIME	CMD
root	4099	0.0	1.0	235772	7808	pts/0	S	15:34	0:00 su jaysimman
jaysimm+	4100	0.0	0.7	224116	5376	pts/0	S	15:34	0:00 bash
root	4130	0.0	0.0	0	0	?	I	15:37	0:00 [kworker/0:3-ata_sff]
jaysimm+	4134	0.0	0.4	225496	3328	pts/0	R+	15:38	0:00 ps aux
jaysimm+	4135	0.0	0.2	220984	1664	pts/0	S+	15:38	0:00 tail -5

```
[jaysimman@vmware ~]$
```

e. ps -u username

i. ps -u root

```
[jaysimman@vmware ~]$ ps -u root | head -5
```

PID	TTY	TIME	CMD
1	?	00:00:13	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	pool_workqueue_
4	?	00:00:00	kworker/R-rcu_g

ii. ps -u yourname

```
[jaysimman@vmware ~]$ ps -u jaysimman | head -5
```

PID	TTY	TIME	CMD
2160	?	00:00:03	systemd
2162	?	00:00:00	(sd-pam)
2177	?	00:00:00	gnome-keyring-d
2181	tty2	00:00:00	gdm-wayland-ses

2. find the pid of bash

a. pidof bash

b. run ps -f (pidofbash)

c. pgrep bash / pgrep -l bash

```
[jaysimman@vmware ~]$ pidof bash
3894
[jaysimman@vmware ~]$ ps -f 3894
```

UID	PID	PPID	C	STIME	TTY	STAT	TIME	CMD
jaysimm+	3894	3893	0	15:07	pts/0	Ss	0:00	-bash

```
[jaysimman@vmware ~]$ pgrep bash
3894
[jaysimman@vmware ~]$ pgrep -l bash
3894 bash
[jaysimman@vmware ~]$ |
```

3. run cat > processfile

a. input some text and suspend or stop the process using ctrl +z

```
[jaysimman@vmware ~]$ cat > jai
Hi
Hello
Bye^Z
[1]+  Stopped                  cat > jai
[jaysimman@vmware ~]$ |
```

b. get the pid of cat using pidof or pgrep cmd

```
[jaysimman@vmware ~]$ pidof cat
4146 4032
[jaysimman@vmware ~]$ pgrep cat
4032
4146
[jaysimman@vmware ~]$ |
```

c. list the process details using ps -f (pidofcat)

```
[jaysimman@vmware ~]$ ps -f 4032
UID          PID    PPID  C STIME TTY          STAT       TIME CMD
jaysimm+    4032    3894  0 15:26 pts/0      T           0:00 cat
[jaysimman@vmware ~]$ |
```

d. Now start the cat process using fg cmd

```
[jaysimman@vmware ~]$ fg
cat > jai
```

e. Check bg jobs if any using jobs cmd.

```
[jaysimman@vmware ~]$ bg
[1]+ cat > jai &

[1]+  Stopped                  cat > jai
[jaysimman@vmware ~]$ |
```

4. Use vi to create a little text file. Suspend vi in background.

```
[jaysimman@vmware ~]$ vi iaj

[2]+  Stopped                  vi iaj
[jaysimman@vmware ~]$ |
```

5. Verify with jobs that vi is suspended in background.

```
[jaysimman@vmware ~]$ bg
[2]+ vi iaj &
[jaysimman@vmware ~]$ |
```

6. Start sleep 100 process, suspend or terminate before it finishes

a. Get the details of sleep process using ps -f pidsleep

```
[jaysimman@vmware ~]$ ps -f 4165
UID          PID    PPID  C STIME TTY          STAT       TIME CMD
jaysimm+    4165     4100  0 15:50 pts/0        S           0:00 sleep 100
[jaysimman@vmware ~]$ |
```

b. Start the sleep cmd in bg.

```
[jaysimman@vmware ~]$ sleep 100 &
[1] 4185
[jaysimman@vmware ~]$ bg
bash: bg: job 1 already in background
[jaysimman@vmware ~]$ |
```

7. Start two long sleep processes in background.

```
[jaysimman@vmware ~]$ sleep 200 &
[2] 4187
[jaysimman@vmware ~]$ sleep 300 &
[3] 4189
```

8. Display all jobs in background

```
[jaysimman@vmware ~]$ jobs
[1]    Running                  sleep 100 &
[2]-   Running                  sleep 200 &
[3]+   Running                  sleep 300 &
```

9. Use pstree cmd to lists all process of bash – pstree pidofbash

```
[jaysimman@vmware ~]$ pidof bash
4372 4329
[jaysimman@vmware ~]$ pstree 4732
[jaysimman@vmware ~]$ pstree 4329
bash
[jaysimman@vmware ~]$ |
```

10. Customise the output columns of ps cmd using

a. ps -eo user,uid,pcpu,pmem,cmd

```
[jaysimman@vmware ~]$ ps -eo user,uid,pcpu,pmem,cmd
USER      UID %CPU %MEM CMD
root      0  0.0  1.1 /usr/lib/systemd/systemd rhgb --switched-root --system --deserialize 31
root      0  0.0  0.0 [kthreadd]
root      0  0.0  0.0 [pool_workqueue_]
root      0  0.0  0.0 [kworker/R-rcu_g]
root      0  0.0  0.0 [kworker/R-rcu_p]
root      0  0.0  0.0 [kworker/R-slab_]
root      0  0.0  0.0 [kworker/R-netns]
root      0  0.0  0.0 [kworker/0:0H-xfs-log/dm-0]
root      0  0.0  0.0 [kworker/R-mm_pe]
root      0  0.0  0.0 [rcu_tasks_kthre]
root      0  0.0  0.0 [rcu_tasks_rude_]
root      0  0.0  0.0 [rcu_tasks_trace]
root      0  0.0  0.0 [ksoftirqd/0]
root      0  0.0  0.0 [rcu_preempt]
root      0  0.0  0.0 [migration/0]
root      0  0.0  0.0 [idle_inject/0]
root      0  0.0  0.0 [cpuhp/0]
root      0  0.0  0.0 [kdevtmpfs]
root      0  0.0  0.0 [kworker/R-inet_]
root      0  0.0  0.0 [kauditd]
root      0  0.0  0.0 [khungtaskd]
root      0  0.0  0.0 [oom_reaper]
root      0  0.0  0.0 [kworker/R-write]
root      0  0.0  0.0 [kcompactd0]
root      0  0.0  0.0 [ksmd]
root      0  0.0  0.0 [khugepaged]
root      0  0.0  0.0 [kworker/R-crypt]
root      0  0.0  0.0 [kworker/R-kinte]
root      0  0.0  0.0 [kworker/R-kbloc]
root      0  0.0  0.0 [kworker/R-blkcg]
root      0  0.0  0.0 [irq/9-acpi]
root      0  0.0  0.0 [kworker/R-tpm_d]
root      0  0.0  0.0 [kworker/R-md]
```

b. ps -eo user=username,uid=useruid,pcpu=cpu,pmem=mem,cmd=command

```
[jaysimman@vmware ~]$ ps -eo user=username,uid=useruid
username useruid
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
root      0
```

c. ps axo user,uid,cmd,stat,pid,ppid

```
[jaysimman@vmware ~]$ ps axo user,uid,cmd,stat,pid,ppid
USER      UID  CMD                                STAT      PID      PPID
root       0    /usr/lib/systemd/systemd rh  Ss        1        0
root       0    [kthreadd]                        S          2        0
root       0    [pool_workqueue_]                S          3        2
root       0    [kworker/R-rcu_g]                 I<         4        2
root       0    [kworker/R-rcu_p]                 I<         5        2
root       0    [kworker/R-slab_]                 I<         6        2
root       0    [kworker/R-netns]                 I<         7        2
root       0    [kworker/0:0H-xfs-log/dm-0]      I<         9        2
root       0    [kworker/R-mm_pe]                 I<        11        2
root       0    [rcu_tasks_kthre]                 I        13        2
root       0    [rcu_tasks_rude_]                 I        14        2
root       0    [rcu_tasks_trace]                 I        15        2
root       0    [ksoftirqd/0]                     S        16        2
```

11. Put one of the sleep process in foreground

```
[jaysimman@vmware ~]$ sleep 50
^Z
[1]+  Stopped                  sleep 50
[jaysimman@vmware ~]$ fg
sleep 50
```

12. Kill one of the sleep process

a. Use job id to kill sleep process – kill %[jobid]

```
[jaysimman@vmware ~]$ kill %1
[1]+  Terminated              sleep 500
[jaysimman@vmware ~]$ |
```

b. Use pid of sleep – kill pid

```
[jaysimman@vmware ~]$ kill -9 4417
[1]+  Killed                    sleep 500
```

13. Kill the vi process using pkill cmd

```
[root@vmware ~]# pkill vi
[root@vmware ~]# |
```

14. Use killall to kill all sleep process

```
[root@vmware ~]# killall sleep
[1]+  Terminated                  sleep 100
[root@vmware ~]# |
```

15. Use kill -9 to kill the bash process

```
[root@vmware ~]# kill -9 4490 4372 4329
Killed
[jaysimman@vmware ~]$ |
```

16. Open 2 terminals and start 2 long sleep processes. (provide screen shots for grep cmd alone in both cases)

a. Put them in bg

```
Welcome Jai
[jaysimman@vmware ~]$ sleep 600 &
[1] 4732
[jaysimman@vmware ~]$ bg
bash: bg: job 1 already in background
[jaysimman@vmware ~]$

[jaysimman@vmware ~]$ sleep 500 &
[1] 4727
[jaysimman@vmware ~]$ bg
bash: bg: job 1 already in background
[jaysimman@vmware ~]$
```

b. Check the jobs in bg

```
[jaysimman@vmware ~]$ jobs
[1]+  Running                  sleep 600 &
[jaysimman@vmware ~]$

[jaysimman@vmware ~]$ jobs
[1]+  Running                  sleep 500 &
[jaysimman@vmware ~]$
```

c. In 1st terminal run -> ps -ef | grep sleep and observe

```
[jaysimman@vmware ~]$ ps -ef | grep sleep
jaysimm+  4727      4329    0 16:47 pts/0    00:00:00 sleep 500
jaysimm+  4732      4698    0 16:47 pts/1    00:00:00 sleep 600
jaysimm+  4774      4698    0 16:50 pts/1    00:00:00 grep --color=auto sleep
[jaysimman@vmware ~]$
```

d. Now close the second terminal and run grep cmd again to check and understand.

```
[jaysimman@vmware ~]$ ps -ef | grep sleep
jaysimm+  4732      4698    0 16:47 pts/1    00:00:00 sleep 600
jaysimm+  4782      4698    0 16:50 pts/1    00:00:00 grep --color=auto sleep
[jaysimman@vmware ~]$
```

e. Inorder to run a process without a terminal we use nohup (hangup cmd)

i. Start sleep process with nohup sleep 100 &

```
[jaysimman@vmware ~]$ nohup sleep 100 &  
[3] 4818  
nohup: ignoring input and appending output to 'nohup.out'
```

ii. In 1st terminal run -> ps -ef | grep sleep and observe

```
[jaysimman@vmware ~]$ ps -ef | grep sleep  
jaysimm+ 4732 4698 0 16:47 pts/1 00:00:00 sleep 600  
jaysimm+ 4818 4698 0 16:53 pts/1 00:00:00 sleep 100  
jaysimm+ 4855 2160 0 16:54 ? 00:00:00 sleep 100  
jaysimm+ 4861 4698 0 16:54 pts/1 00:00:00 grep --color=auto sleep  
[jaysimman@vmware ~]$
```

iii. Now close the second terminal and run grep cmd again to check and understand.

```
[jaysimman@vmware ~]$ ps -ef | grep sleep  
jaysimm+ 4732 4698 0 16:47 pts/1 00:00:00 sleep 600  
jaysimm+ 4868 4698 0 16:57 pts/1 00:00:00 grep --color=auto sleep  
[3]+ Done nohup sleep 100
```

f. Observe a new file nohup.out is created cat and ch

```
[jaysimman@vmware ~]$ cat nohup.out  
[1]+ Done sleep 600  
[jaysimman@vmware ~]$
```