1. Creating a process:

```
ubuntu@ubuntu-OptiPlex-SFF-7020:~$ cd sanjana
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ nano process.sh
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ bash process.sh
Parent PID: 7526
Child process created
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

Nano shellscript input:

```
  GNU nano 7.2                          process.sh
#!/bin/bash
echo "Parent PID: $$"
sleep 5 &
echo "Child process created"
```

2. Ps - show running process
   - ps -f
   - ps -ef

```
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps
    PID TTY          TIME CMD
   7269 pts/0    00:00:00 bash
   8922 pts/0    00:00:00 ps
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
buntu       7269    7261  0 14:06 pts/0    00:00:00 bash
buntu       8924    7269  0 14:17 pts/0    00:00:00 ps -f
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
oot            1       0  0 13:51 ?        00:00:02 /sbin/init splash
oot            2       0  0 13:51 ?        00:00:00 [kthreadd]
oot            3       2  0 13:51 ?        00:00:00 [pool_workqueue_release]
oot            4       2  0 13:51 ?        00:00:00 [kworker/R-rcu_gp]
oot            5       2  0 13:51 ?        00:00:00 [kworker/R-sync_wq]
oot            6       2  0 13:51 ?        00:00:00 [kworker/R-kvfree_rcu_reclai
oot            7       2  0 13:51 ?        00:00:00 [kworker/R-slub_flushwq]
oot            8       2  0 13:51 ?        00:00:00 [kworker/R-netns]
```

3. top - live process monitoring

```
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ top

op - 14:20:58 up 29 min,  1 user,  load average: 0.54, 0.44, 0.34
asks: 384 total,   1 running, 383 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
iB Mem :  15675.7 total,   9038.8 free,   3213.7 used,   4445.4 buff/cache
iB Swap:   4096.0 total,   4096.0 free,      0.0 used.  12462.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0   23528  14548   9316 S   0.0   0.1   0:02.16 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_wo+
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.11 kworker+
     12 root      20   0       0      0      0 I   0.0   0.0   0:03.83 kworker+
     13 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.08 ksoftir+
     15 root      20   0       0      0      0 I   0.0   0.0   0:02.99 rcu_pre+
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_exp+
```

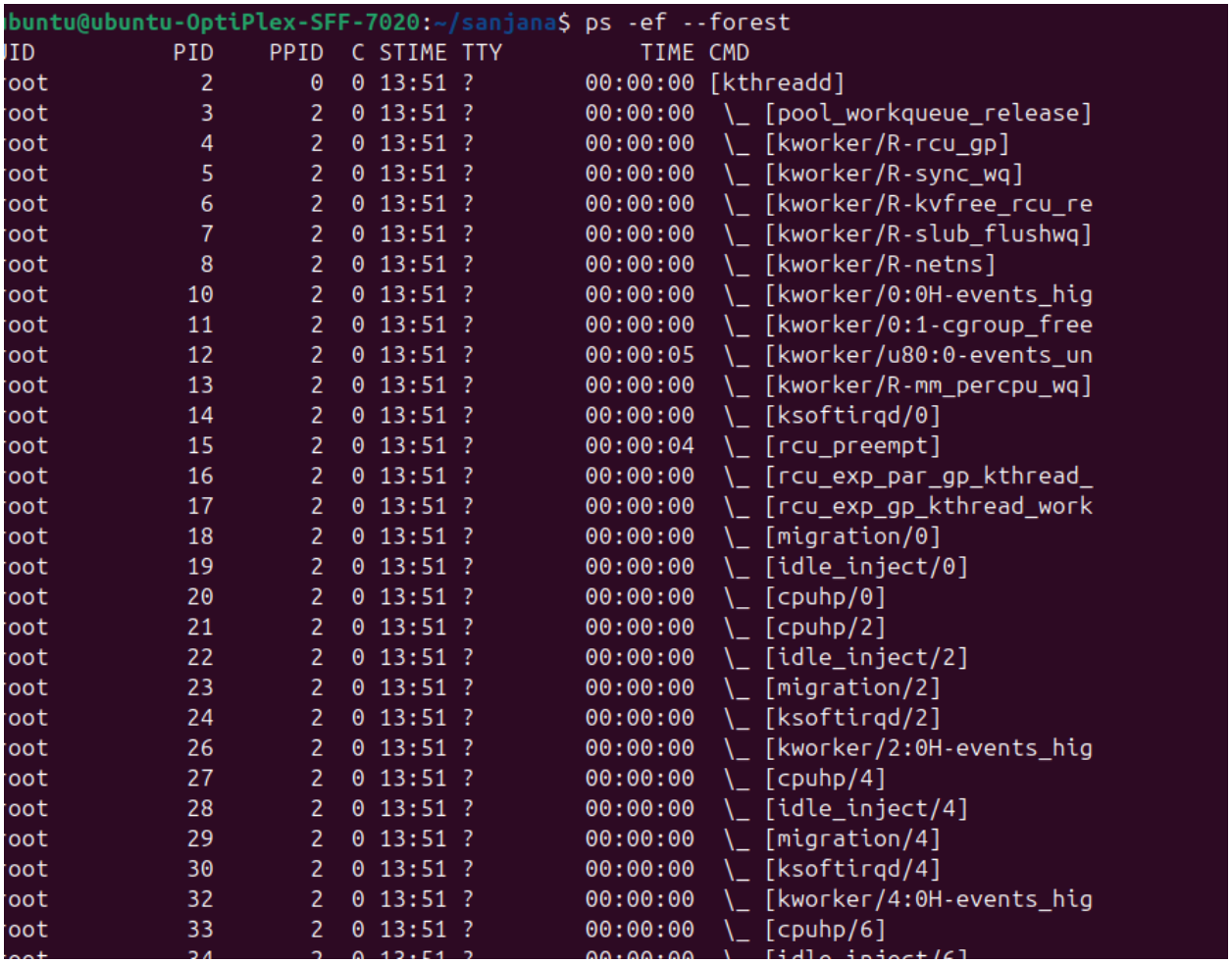
4. Jobs: show background jobs
   Fg - bring job to foreground
   Bg - resume background jobs

```
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ sleep 5&
1] 9579
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ jobs
1]+  Running                 sleep 5 &
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

```
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ bg %1
ash: bg: job has terminated
1]+  Done                    sleep 5
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ fg %1
ash: fg: %1: no such job
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

5. Kill : Terminate process
   kill PID
   kill -9 PID

6. Display Parent-child relationship
   ps -ef --forest
   Or
   pstree : shows the relation of parent and child commands



```
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps -ef --forest
UID          PID    PPID  C STIME TTY          TIME CMD
oot            2       0  0 13:51 ?        00:00:00 [kthreadd]
oot            3       2  0 13:51 ?        00:00:00  \_ [pool_workqueue_release]
oot            4       2  0 13:51 ?        00:00:00  \_ [kworker/R-rcu_gp]
oot            5       2  0 13:51 ?        00:00:00  \_ [kworker/R-sync_wq]
oot            6       2  0 13:51 ?        00:00:00  \_ [kworker/R-kvfree_rcu_re
oot            7       2  0 13:51 ?        00:00:00  \_ [kworker/R-slub_flushwq]
oot            8       2  0 13:51 ?        00:00:00  \_ [kworker/R-netns]
oot           10       2  0 13:51 ?        00:00:00  \_ [kworker/0:0H-events_hig
oot           11       2  0 13:51 ?        00:00:00  \_ [kworker/0:1-cgroup_free
oot           12       2  0 13:51 ?        00:00:05  \_ [kworker/u80:0-events_un
oot           13       2  0 13:51 ?        00:00:00  \_ [kworker/R-mm_percpu_wq]
oot           14       2  0 13:51 ?        00:00:00  \_ [ksoftirqd/0]
oot           15       2  0 13:51 ?        00:00:04  \_ [rcu_preempt]
oot           16       2  0 13:51 ?        00:00:00  \_ [rcu_exp_par_gp_kthread_
oot           17       2  0 13:51 ?        00:00:00  \_ [rcu_exp_gp_kthread_work
oot           18       2  0 13:51 ?        00:00:00  \_ [migration/0]
oot           19       2  0 13:51 ?        00:00:00  \_ [idle_inject/0]
oot           20       2  0 13:51 ?        00:00:00  \_ [cpuhp/0]
oot           21       2  0 13:51 ?        00:00:00  \_ [cpuhp/2]
oot           22       2  0 13:51 ?        00:00:00  \_ [idle_inject/2]
oot           23       2  0 13:51 ?        00:00:00  \_ [migration/2]
oot           24       2  0 13:51 ?        00:00:00  \_ [ksoftirqd/2]
oot           26       2  0 13:51 ?        00:00:00  \_ [kworker/2:0H-events_hig
oot           27       2  0 13:51 ?        00:00:00  \_ [cpuhp/4]
oot           28       2  0 13:51 ?        00:00:00  \_ [idle_inject/4]
oot           29       2  0 13:51 ?        00:00:00  \_ [migration/4]
oot           30       2  0 13:51 ?        00:00:00  \_ [ksoftirqd/4]
oot           32       2  0 13:51 ?        00:00:00  \_ [kworker/4:0H-events_hig
oot           33       2  0 13:51 ?        00:00:00  \_ [cpuhp/6]
```

```
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ pstree
systemd─┬─ModemManager───3*[{ModemManager}]
        ├─NetworkManager───3*[{NetworkManager}]
        ├─accounts-daemon───3*[{accounts-daemon}]
        ├─anacron
        ├─avahi-daemon───avahi-daemon
        ├─bluetoothd
        ├─colord───3*[{colord}]
        ├─cron
        ├─cups-browsed───3*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon
        ├─fwupd───5*[{fwupd}]
        ├─gdm3─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b───3*[{gnome-session-b}]
        │      │                 │                 └─3*[{gdm-wayland-ses}]
        │      │                 └─3*[{gdm-session-wor}]
        │      └─3*[{gdm3}]
        ├─gnome-remote-de───3*[{gnome-remote-de}]
        ├─2*[kerneloops]
        ├─polkitd───3*[{polkitd}]
        ├─power-profiles-───3*[{power-profiles-}]
        ├─rsyslogd───3*[{rsyslogd}]
        ├─rtkit-daemon───2*[{rtkit-daemon}]
        ├─snapd───26*[{snapd}]
        ├─switcheroo-cont───3*[{switcheroo-cont}]
        ├─systemd─┬─(sd-pam)
        │         ├─at-spi2-registr───3*[{at-spi2-registr}]
        │         ├─crashhelper───{crashhelper}
        │         ├─dbus-daemon
        │         ├─dconf-service───3*[{dconf-service}]
```

7. Creating Multiple Child Process:
   Creating 3 child process in nano shellscript
   Output:

```
Parent PID:10555
UID          PID    PPID  C STIME TTY          TIME CMD
ubuntu      7269    7261  0 14:06 pts/0    00:00:00 bash
ubuntu     10555    7269  0 14:48 pts/0    00:00:00  \_ bash smth.sh
ubuntu     10556   10555  0 14:48 pts/0    00:00:00      \_ sleep 30
ubuntu     10557   10555  0 14:48 pts/0    00:00:00      \_ sleep 40
ubuntu     10558   10555  0 14:48 pts/0    00:00:00      \_ sleep 50
ubuntu     10559   10555  0 14:48 pts/0    00:00:00      \_ ps -f --forest
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

Input of the shellscript:

```
 GNU nano 7.2                                                    smth.sh
#!/bin/bash
echo "Parent PID:$$"
sleep 30 &
sleep 40 &
sleep 50 &
ps -f --forest
```

Step 1: To observe PID and PPID of sleep:
Write:

```
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ nano smth.sh
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ echo $$
7269
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ubuntu      7269    7261  0 14:06 pts/0    00:00:00 bash
ubuntu     10996    7269  0 14:54 pts/0    00:00:00 ps -f
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ sleep 30
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

Do this in another terminal:

```
u@ubuntu-OptiPlex-SFF-7020:~$ ps -ef | grep sleep
u      10998    7269  0 14:54 pts/0    00:00:00 sleep 30
u      11030   11011  0 14:54 pts/1    00:00:00 grep --color=auto sleep
u@ubuntu-OptiPlex-SFF-7020:~$
```

PPID will match the PID of your shell (7269)

Step 3: Create a Background Child Process
Sleep 60 &

```
^C
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ sleep 60 &
[1] 12011
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

[1] = Job Number
12011 = PID of job process
Checking jobs:

```
[1] 12011
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ jobs
[1]+  Done                    sleep 60
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

Step 4: Displaying Parent Child Relations using ps
Shows:

- UID
- PID
- PPID
- CMD

```
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ubuntu      7269    7261  0 14:06 pts/0    00:00:00 bash
ubuntu     12554    7269  0 15:13 pts/0    00:00:00 ps -f
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

Using Forest Format (Best Method)
ps -ef --forest

```
buntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps -ef --forest
ID          PID    PPID  C STIME TTY          TIME CMD
oot           2       0  0 13:51 ?        00:00:00 [kthreadd]
oot           3       2  0 13:51 ?        00:00:00  \_ [pool_workqueue_release]
oot           4       2  0 13:51 ?        00:00:00  \_ [kworker/R-rcu_gp]
oot           5       2  0 13:51 ?        00:00:00  \_ [kworker/R-sync_wq]
oot           6       2  0 13:51 ?        00:00:00  \_ [kworker/R-kvfree_rcu_re
oot           7       2  0 13:51 ?        00:00:00  \_ [kworker/R-slub_flushwq]
oot           8       2  0 13:51 ?        00:00:00  \_ [kworker/R-netns]
oot          10       2  0 13:51 ?        00:00:00  \_ [kworker/0:0H-events_hig
oot          11       2  0 13:51 ?        00:00:00  \_ [kworker/0:1-cgroup_free
oot          12       2  0 13:51 ?        00:00:05  \_ [kworker/u80:0-events_un
oot          13       2  0 13:51 ?        00:00:00  \_ [kworker/R-mm_percpu_wq]
oot          14       2  0 13:51 ?        00:00:00  \_ [ksoftirqd/0]
oot          15       2  0 13:51 ?        00:00:04  \_ [rcu_preempt]
oot          16       2  0 13:51 ?        00:00:00  \_ [rcu_exp_par_gp_kthread_
oot          17       2  0 13:51 ?        00:00:00  \_ [rcu_exp_gp_kthread_work
oot          18       2  0 13:51 ?        00:00:00  \_ [migration/0]
oot          19       2  0 13:51 ?        00:00:00  \_ [idle_inject/0]
oot          20       2  0 13:51 ?        00:00:00  \_ [cpuhp/0]
oot          21       2  0 13:51 ?        00:00:00  \_ [cpuhp/2]
oot          22       2  0 13:51 ?        00:00:00  \_ [idle_inject/2]
oot          23       2  0 13:51 ?        00:00:00  \_ [migration/2]
oot          24       2  0 13:51 ?        00:00:00  \_ [ksoftirqd/2]
oot          26       2  0 13:51 ?        00:00:00  \_ [kworker/2:0H-events_hig
oot          27       2  0 13:51 ?        00:00:00  \_ [cpuhp/4]
oot          28       2  0 13:51 ?        00:00:00  \_ [idle_inject/4]
oot          29       2  0 13:51 ?        00:00:00  \_ [migration/4]
oot          30       2  0 13:51 ?        00:00:00  \_ [ksoftirqd/4]
oot          32       2  0 13:51 ?        00:00:00  \_ [kworker/4:0H-events_hig
oot          33       2  0 13:51 ?        00:00:00  \_ [cpuhp/6]
```

Step 5: Create Multiple Child Process

```
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ sleep 100 &
[1] 12826
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ sleep 120 &
[2] 12828
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ sleep 240 &
[3] 12829
```

```
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ps -f --forest
UID          PID    PPID  C STIME TTY          TIME CMD
ubuntu      7269    7261  0 14:06 pts/0    00:00:00 bash
ubuntu     12826    7269  0 15:15 pts/0    00:00:00  \_ sleep 100
ubuntu     12828    7269  0 15:15 pts/0    00:00:00  \_ sleep 120
ubuntu     12829    7269  0 15:15 pts/0    00:00:00  \_ sleep 240
ubuntu     12850    7269  0 15:16 pts/0    00:00:00  \_ ps -f --forest
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

Step 6: Child Process Creation using shellscript

```
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ nano child.sh
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ chmod +x child.sh
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$ ./child.sh
Parent PID:13576
UID          PID    PPID  C STIME TTY          TIME CMD
ubuntu      7269    7261  0 14:06 pts/0    00:00:00 bash
ubuntu     13576    7269  0 15:25 pts/0    00:00:00  \_ /bin/bash ./child.sh
ubuntu     13577   13576  0 15:25 pts/0    00:00:00      \_ sleep 200
ubuntu     13578   13576  0 15:25 pts/0    00:00:00      \_ sleep 250
ubuntu     13579   13576  0 15:25 pts/0    00:00:00      \_ ps -f --forest
ubuntu@ubuntu-OptiPlex-SFF-7020:~/sanjana$
```

Nano shellscript for input:

```
#!/bin/bash
echo "Parent PID:$$"
sleep 200 &
sleep 250 &
ps -f --forest
```