Process Creation -

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

```
rvu@rvu-OptiPlex-SFF-7020:~$ nano process.sh
rvu@rvu-OptiPlex-SFF-7020:~$ bash process.sh
process.sh: line 2: echoParent PID: 4159: command not found
process.sh: line 4: echoChild Process Control: command not found
rvu@rvu-OptiPlex-SFF-7020:~$ nano process.sh
rvu@rvu-OptiPlex-SFF-7020:~$ bash process.sh
Parent PID: 4176
Child Process Control
rvu@rvu-OptiPlex-SFF-7020:~$ ps
    PID TTY          TIME CMD
   3190 pts/0    00:00:00 bash
   4182 pts/0    00:00:00 ps
rvu@rvu-OptiPlex-SFF-7020:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
rvu         3190    3183  0 19:36 pts/0    00:00:00 bash
rvu         4183    3190  0 19:48 pts/0    00:00:00 ps -f
rvu@rvu-OptiPlex-SFF-7020:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 19:29 ?        00:00:02 /sbin/init splash
root           2       0  0 19:29 ?        00:00:00 [kthreadd]
root           3       2  0 19:29 ?        00:00:00 [pool_workqueue_release]
root           4       2  0 19:29 ?        00:00:00 [kworker/R-rcu_gp]
root           5       2  0 19:29 ?        00:00:00 [kworker/R-sync_wq]
root           6       2  0 19:29 ?        00:00:00 [kworker/R-kvfree_rcu_reclaim]
root           7       2  0 19:29 ?        00:00:00 [kworker/R-slub_flushwq]
root           8       2  0 19:29 ?        00:00:00 [kworker/R-netns]
root          10       2  0 19:29 ?        00:00:00 [kworker/0:0H-events_highpri]
root          11       2  0 19:29 ?        00:00:00 [kworker/0:1-events]
root          13       2  0 19:29 ?        00:00:00 [kworker/R-mm_percpu_wq]
root          14       2  0 19:29 ?        00:00:00 [rcu_tasks_kthread]
root          15       2  0 19:29 ?        00:00:00 [rcu_tasks_rude_kthread]
root          16       2  0 19:29 ?        00:00:00 [rcu_tasks_trace_kthread]
root          17       2  0 19:29 ?        00:00:00 [ksoftirqd/0]
root          18       2  0 19:29 ?        00:00:00 [rcu_preempt]
root          19       2  0 19:29 ?        00:00:00 [rcu_exp_par_gp_kthread_worker/1]
root          20       2  0 19:29 ?        00:00:00 [rcu_exp_gp_kthread_worker]
root          21       2  0 19:29 ?        00:00:00 [migration/0]
```

1. Nano process.sh is used to edit and make the changes in terminal

Ps stands for “process status”, this command is a snapshot of a complete running process in the system

Ps -f displays a “full” format listing of your processes

Ps -ef is used as a common combination of simple process management in bash

Nano process.sh : This program provides a description of simple process management in bash

```
rvu@rvu-OptiPlex-SFF-7020:~$ top

top - 19:48:56 up 19 min,  1 user,  load average: 0.00, 0.04, 0.03
Tasks: 355 total,   1 running, 354 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.2 sy,  0.0 ni, 99.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15677.3 total,  12845.0 free,   1524.4 used,   1818.2 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.  14152.9 avail Mem

    PID     USER  PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2467      rvu  20   0 5039260 279068 124708 S   4.6   1.7   0:13.57 gnome-shell
   3183      rvu  20   0  557848  55944  44620 S   2.0   0.3   0:02.83 gnome-terminal-
    192     root   0 -20       0      0      0 I   0.7   0.0   0:00.74 kworker/u81:0-i915_flip
     18     root  20   0       0      0      0 I   0.3   0.0   0:00.53 rcu_preempt
    554     root  20   0       0      0      0 I   0.3   0.0   0:00.45 kworker/u80:8-flush-259:0
   4185      rvu  20   0   14528   5748   3668 R   0.3   0.0   0:00.12 top
      1     root  20   0   23596  13972   9332 S   0.0   0.1   0:02.13 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_workqueue_release
      4     root   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_gp
      5     root   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-sync_wq
      6     root   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-kvfree_rcu_reclaim
      7     root   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_flushwq
      8     root   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
     10     root   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     11     root  20   0       0      0      0 I   0.0   0.0   0:00.15 kworker/0:1-events
     13     root   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_percpu_wq
     14     root  20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
     15     root  20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     16     root  20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     17     root  20   0       0      0      0 S   0.0   0.0   0:00.01 ksoftirqd/0
     19     root  20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_exp_par_gp_kthread_worker/1
     20     root  20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_exp_gp_kthread_worker
     21     root  rt   0       0      0      0 S   0.0   0.0   0:00.02 migration/0
     22     root -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
```

2. Top provides a real time dynamic view of running process

```
rvu@rvu-OptiPlex-SFF-7020:~$ jobs
rvu@rvu-OptiPlex-SFF-7020:~$ bg%1
bg%1: command not found
rvu@rvu-OptiPlex-SFF-7020:~$ bg % 1
bash: bg: %: no such job
bash: bg: 1: no such job
```

3. bg % 1 resumes a stopped job

Jobs lists all active jobs

fg % 1 brings a background job to the foreground

```
rvu@rvu-OptiPlex-SFF-7020:~$ pstree
systemd─┬─ModemManager───3*[{ModemManager}]
        ├─NetworkManager───3*[{NetworkManager}]
        ├─accounts-daemon───3*[{accounts-daemon}]
        ├─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───25*[{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}]
        │         ├─evolution-addre───6*[{evolution-addre}]
        │         ├─evolution-calen───9*[{evolution-calen}]
        │         ├─evolution-sourc───4*[{evolution-sourc}]
        │         ├─gcr-ssh-agent───2*[{gcr-ssh-agent}]
        │         ├─2*[gjs───11*[{gjs}]]
        │         ├─gnome-keyring-d───4*[{gnome-keyring-d}]
```

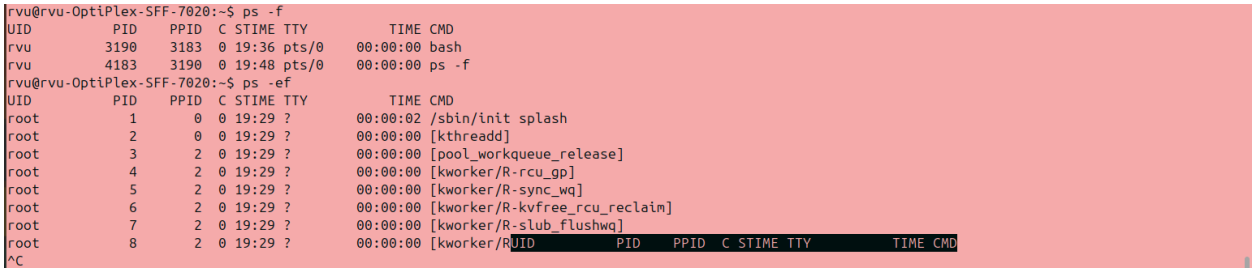
4. Pstree shows the running processes as a tree

```
rvu@rvu-OptiPlex-SFF-7020: ~                    rvu@rvu-OptiPlex-SFF-7020: ~
  GNU nano 7.2                                  process.sh
#!/bin/bash
echo "Parent PID: $$"
sleep 30 &
sleep 40 &
sleep 50 &
ps -f --forest
```

5. This script is a demonstration of process management and parent-child relationships in a LInux environment

Output -

```
rvu@rvu-OptiPlex-SFF-7020:~$ nano process.sh
rvu@rvu-OptiPlex-SFF-7020:~$ bash process.sh
Parent PID: 8896
UID          PID    PPID  C STIME TTY          TIME CMD
rvu         3190    3183  0 14:06 pts/0    00:00:00 bash
rvu         8896    3190  0 14:50 pts/0    00:00:00  \_ bash process.sh
rvu         8897    8896  0 14:50 pts/0    00:00:00      \_ sleep 30
rvu         8898    8896  0 14:50 pts/0    00:00:00      \_ sleep 40
rvu         8899    8896  0 14:50 pts/0    00:00:00      \_ sleep 50
rvu         8900    8896  0 14:50 pts/0    00:00:00      \_ ps -f --forest
```



6. Ps -f is used to see the parent-child relationship

ps -ef is used to provide a complete tree structure