

1. Ejecuta la orden ps para descubrir los procesos en curso. Crea un nuevo subshell con la orden sh y descubre su presencia ejecutando de nuevo ps.

```
danielc@danielc-VirtualBox:~$ ps
  PID TTY          TIME CMD
  2722 pts/0        00:00:00 bash
  3278 pts/0        00:00:00 ps
danielc@danielc-VirtualBox:~$ sh
$ ps
  PID TTY          TIME CMD
  2722 pts/0        00:00:00 bash
  3279 pts/0        00:00:00 sh
  3280 pts/0        00:00:00 ps
```

2. Visualiza los distintos formatos de ps ejecutando la orden con las opciones f, l, e, a, el sucesivamente.

```
danielc@danielc-VirtualBox:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
danielc      2722    2676  0 19:28 pts/0        00:00:00 bash
danielc      3363    2722  0 19:35 pts/0        00:00:00 ps -f
danielc      1342     272  0 00:00 tty2        00:00:01 Xorg
danielc      1378     272  0 00:00 tty2        00:00:00 gnome-session-b
danielc      3384     272  0 00:00 pts/0        00:00:00 ps
```

```
danielc@danielc-VirtualBox:~$ ps -l
F S      UID          PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S      1000         2722    2676  0  80   0  -   3289 do_wai pts/0        00:00:00 bash
0 R      1000         3376    2722  0  80   0  -   3487 -          pts/0        00:00:00 ps
```

```
danielc@danielc-VirtualBox:~$ ps -e
  PID TTY          TIME CMD
    1 ?           00:00:00 systemd
    2 ?           00:00:00 kthreadd
    3 ?           00:00:00 rcu_gp
    4 ?           00:00:00 rcu_par_gp
    6 ?           00:00:00 kworker/0:0H-events_highpri
    8 ?           00:00:00 kworker/u2:0-events_power_efficient
    9 ?           00:00:00 mm_percpu_wq
   10 ?           00:00:00 rcu_tasks_rude_
   11 ?           00:00:00 rcu_tasks_trace
```

3. Ejecuta la orden ps y comprueba su PID como proceso. Ejecútala de nuevo y comprueba como su PID se ha incrementado. ¿Se ha incrementado también su PPID?

```
danielc@danielc-VirtualBox:~$ ps -l
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   1000     2722    2676  0  80   0  -  3289 do_wai pts/0        00:00:00 bash
0 R   1000     3388    2722  0  80   0  -  3487 -          pts/0        00:00:00 ps
danielc@danielc-VirtualBox:~$ ps -l
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   1000     2722    2676  0  80   0  -  3289 do_wai pts/0        00:00:00 bash
0 R   1000     3389    2722  0  80   0  -  3487 -          pts/0        00:00:00 ps
danielc@danielc-VirtualBox:~$
```

No se ha incrementado, por lo que el proceso padre sigue siendo el mismo.

4. Ejecuta varias ordenes cortas (who, clear, date, etc.) y luego vuelve a ejecutar la orden ps para comprobar cuanto ha aumentado el PID.

```
danielc@danielc-VirtualBox:~$ who
danielc  :0                2022-05-06 19:27 (:0)
danielc@danielc-VirtualBox:~$ ps
  PID TTY          TIME CMD
  2722 pts/0        00:00 bash
  3399 pts/0        00:00 ps
```

5. ¿Cuántos procesos se han ejecutado en el sistema desde tu incorporación al sistema?

Ya que el PID del comando ps es 3399, se habrán ejecutado 3399 procesos desde que se inició el sistema.

6. Visualiza los procesos asociados a tu terminal. Ejecuta a continuación la ordenes `ps -el` y detecta en la lista los procesos de tu terminal obtenidos en la acción anterior.

```
danielc@danielc-VirtualBox:~$ ps -a
  PID TTY          TIME CMD
 1342 tty2        00:00:03 Xorg
 1378 tty2        00:00:00 gnome-session-b
 3469 pts/0        00:00:00 ps
```

```
0 S 1000 1338 1274 0 80 0 - 137511 poll_s ? 00:00:00 goa-daem
4 S 1000 1340 1261 0 80 0 - 41637 poll_s tty2 00:00:00 gdm-x-se
4 S 1000 1342 1340 0 80 0 - 64545 ep_pol tty2 00:00:03 Xorg
0 S 1000 1349 1274 0 80 0 - 98718 poll_s ? 00:00:00 goa-iden
0 S 1000 1351 1274 0 80 0 - 60128 poll_s ? 00:00:00 gvfs-gph
0 S 1000 1378 1340 0 80 0 - 48316 poll_s tty2 00:00:00 gnome-se

0 S 1000 1349 1274 0 80 0 - 98718 poll_s ? 00:00:00 goa-ident
0 S 1000 1351 1274 0 80 0 - 60128 poll_s ? 00:00:00 gvfs-gpho
0 S 1000 1378 1340 0 80 0 - 48316 poll_s tty2 00:00:00 gnome-ses
1 S 1000 1447 1378 0 80 0 - 1510 - ? 00:00:00 ssh-agent
0 S 1000 1466 1274 0 80 0 - 95885 poll_s ? 00:00:00 at-spi-bu
0 S 1000 1473 1466 0 80 0 - 4844 - poll_s ? 00:00:00 dbus-daem

0 S 1000 3057 1486 0 80 0 - 124771 poll_s ? 00:00:00 update-no
0 S 1000 3214 1274 0 90 10 - 231315 poll_s ? 00:00:06 update-ma
1 I 0 3401 2 0 80 0 - 0 - ? 00:00:00 kworker/u
1 I 0 3464 2 0 80 0 - 0 - ? 00:00:00 kworker/0
0 R 1000 3470 2722 0 80 0 - 3487 - pts/0 00:00:00 ps
```

Y claro, el proceso `ps` tiene un PID mayor ya que he ejecutado el comando `ps -el`.

7. Comprueba los procesos que no están asociados a ningún terminal en concreto. Visualizando con la opción `-ef` las ordenes que generan dichos procesos, intenta comprender por que en el campo correspondiente aparece el carácter?

El carácter '?' indicado que el proceso es un proceso del sistema.

8. Construye el árbol de procesos vivos del sistema desde su inicio.

`ps tree`

9. Ejecuta la orden `ps` en segundo plano y anota su PID y su PPID. Visualiza el valor de las variables `$` y `!`. ¿Qué relación hay?

```
danielc@danielc-VirtualBox:~$ PID TTY          TIME CMD
 2321 pts/0        00:00:00 bash
 2339 pts/0        00:00:00 ps
echo $! ; echo $$
2339
2321
[1]+  Done                  ps
```

10. ¿Que orden ejecutaría el programa prog en segundo plano?

Prog &

11. Ejecuta la orden ps -el en segundo plano. ¿puedes identificar la propia orden ps – el en el listado? ¿No se ha ejecutado en primer plano a pesar de haber especificado el carácter &?

```
0 S 1000 1900 1451 0 99 - - 150327 do_pol ? 00:00:00 tracker-m
0 S 1000 1922 1451 0 80 0 - 649388 do_pol ? 00:00:00 gjs
0 S 1000 1961 1451 0 80 0 - 40724 do_pol ? 00:00:00 gvfsd-met
1 I 0 1999 2 0 80 0 - 0 - ? 00:00:00 kworker/0
0 S 1000 2005 1538 0 80 0 - 86050 do_pol ? 00:00:00 update-no
0 S 1000 2108 1559 0 80 0 - 697275 do_pol ? 00:00:00 gjs
0 S 1000 2144 1451 0 80 0 - 198652 do_pol ? 00:00:00 gnome-cal
0 S 1000 2148 1451 0 80 0 - 169432 do_pol ? 00:00:00 seahorse
0 S 1000 2152 1451 0 80 0 - 139611 do_pol ? 00:00:00 gnome-ter
1 I 0 2178 2 0 80 0 - 0 - ? 00:00:00 kworker/u
0 S 1000 2321 2152 0 80 0 - 2773 do_sel pts/0 00:00:00 bash
1 I 0 2329 2 0 80 0 - 0 - ? 00:00:00 kworker/0
0 R 1000 2346 2321 0 80 0 - 3124 - pts/0 00:00:00 ps
echo $!
2346
[1]+ Done ps -el
danielc@danielc-VirtualBox:~$
```

12. El archive AGENDA contiene gran cantidad de datos (una línea por cada registro). Pretendemos ordena alfabéticamente sus entradas y guardar la nueva lista ordenada en el archivo AGENDA.ORD ¿Cómo hacerlo en ejecución desatendida ya que el proceso puede ser largo?

sort AGENDA > AGENDA.ORD &

13. La orden find /-name motd –print busca en toda la estructura de archivos del sistema un archivo de nombre motd, comenzando la búsqueda por el directorio raíz y mostrando al final su ruta de acceso. Ejecuta la orden y observa la salida del prompt. Repite la ejecución en segundo plano.

```
danielc@danielc-VirtualBox:~$ find / -name motd -print 1>resultado.txt 2>null
danielc@danielc-VirtualBox:~$ cat resultado.txt
/snap/core20/1434/etc/motd
/snap/core20/1434/etc/writable/motd
/snap/core20/1434/usr/share/base-files/motd
/snap/core20/1405/etc/motd
/snap/core20/1405/etc/writable/motd
/snap/core20/1405/usr/share/base-files/motd
/snap/core/12834/etc/motd
/snap/core/12834/usr/share/base-files/motd
/usr/share/base-files/motd
/usr/share/doc/util-linux/examples/motd
danielc@danielc-VirtualBox:~$
```

14. La orden `sleep 15` provoca una espera de 15 segundos. Ejecútala en segundo plano y aprovecha el descanso para ejecutar la orden `ps -l` y descubrir la entrada correspondiente a dicha orden `sleep`.

```
danielc@danielc-VirtualBox:~$ sleep 15 &
[1] 2508
danielc@danielc-VirtualBox:~$ ps -l
F S    UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S    1000      2321     2152  0  80   0  -  2773 do_wai pts/0        00:00:00 bash
0 S    1000      2508     2321  0  80   0  -  2088 hrtim pts/0        00:00:00 sleep
0 R    1000      2509     2321  0  80   0  -  3124 -      pts/0        00:00:00 ps
danielc@danielc-VirtualBox:~$
```

15. Indica las diferencias que hay entre las ordenes:

`Prog1;prog2;prog3`: Ejecuta `prog1`, `prog2` y `prog3`n

`(prog1; prog2; prog3)&`: Ejecuta `prog1`, `prog2` y `prog3` todos en segundo plano.

`Prog1; prog2; prog3&`: Ejecuta `prog1`, `prog2` y `prog3`, siendo `prog3` en segundo plano.

`(prog1; prog2; prog3) > papelera &`: Ejecuta `prog1`, `prog2` y `prog3` y redirecciona el resultado a la `papelera`, todo esto en segundo plano.

16 . Ejecuta las líneas de órdenes:

`sleep 15; echo he terminado`

`(sleep15; echo he terminado) &`

e intenta en ambos casos ejecutar la orden `ps -f` en el tiempo de espera.

```
danielc@danielc-VirtualBox:~$ sleep 15; echo he terminado
ps -f
he terminado
danielc@danielc-VirtualBox:~$ ps -f
UID      PID     PPID  C STIME TTY          TIME CMD
danielc  2321     2152  0  22:01 pts/0        00:00:00 bash
danielc  2526     2321  0  22:29 pts/0        00:00:00 ps -f
danielc@danielc-VirtualBox:~$ (sleep 15; echo he terminado) &
[1] 2529
danielc@danielc-VirtualBox:~$ ps -f
UID      PID     PPID  C STIME TTY          TIME CMD
danielc  2321     2152  0  22:01 pts/0        00:00:00 bash
danielc  2529     2321  0  22:29 pts/0        00:00:00 bash
danielc  2530     2529  0  22:29 pts/0        00:00:00 sleep 15
danielc  2531     2321  0  22:29 pts/0        00:00:00 ps -f
danielc@danielc-VirtualBox:~$ he terminado
```

17. Crea el siguiente archivos y ejecútalo:

```
cat >SINFÍN
while true
do
echo sigo
sleep 5
done
```

```
danielc@danielc-VirtualBox:~$ cat > sinfin
while true
do
echo sigo
sleep 5
done
```

```
sudo: ./sinfin: command not found
danielc@danielc-VirtualBox:~$ sudo chmod +x sinfin
danielc@danielc-VirtualBox:~$ ./sinfin
sigo
sigo
sigo
sigo
sigo
sigo
sigo
sigo
```

18. Ejecuta el programa de la práctica anterior en segundo plano. Durante los tiempos de espera en los que SINFÍN esta durmiendo invoca varias veces la orde `ps -l` . el número de identificación del proceso PID de la orden `sleep` ¿es siempre el mismo? ¿Por qué?

```
[1] 2009
danielc@danielc-VirtualBox:~$ sigo
ps -l
F S  UID      PID     PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
0 S  1000      2856     2799  0   80   0  -  2740 do_wai pts/1        00:00:00 bash
1 S  1000      2863     2856  0   80   0  -  2740 do_wai pts/1        00:00:00 bash
0 S  1000      2864     2863  0   80   0  -  2088 hrtime pts/1        00:00:00 sleep
0 R  1000      2865     2856  0   80   0  -  3124 -      pts/1        00:00:00 ps
danielc@danielc-VirtualBox:~$ sigo
ps -l
F S  UID      PID     PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
0 S  1000      2856     2799  0   80   0  -  2740 do_wai pts/1        00:00:00 bash
1 S  1000      2863     2856  0   80   0  -  2740 do_wai pts/1        00:00:00 bash
0 S  1000      2867     2863  0   80   0  -  2088 hrtime pts/1        00:00:00 sleep
0 R  1000      2868     2856  0   80   0  -  3124 -      pts/1        00:00:00 ps
danielc@danielc-VirtualBox:~$ sigo
ps -l
F S  UID      PID     PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
0 S  1000      2856     2799  0   80   0  -  2740 do_wai pts/1        00:00:00 bash
1 S  1000      2863     2856  0   80   0  -  2740 do_wai pts/1        00:00:00 bash
0 S  1000      2870     2863  0   80   0  -  2088 hrtime pts/1        00:00:00 sleep
0 R  1000      2871     2856  0   80   0  -  3124 -      pts/1        00:00:00 ps
danielc@danielc-VirtualBox:~$ sigo
```

No es el mismo ya que el comando se ejecuta una vez tras otra en bucle, por lo que su PID cambia.

19. Visualiza los tiempos asociados a la ejecución de las siguientes órdenes:

ls

ls /-R

ps

cat *

```
sys 0m0.000s
danielc@danielc-VirtualBox:~$ time ls 1>null 2>null; time ls / -r 1>null 2>null
; time ps 1>null 2>null; time cat * 1>null 2>null

real    0m0.001s
user    0m0.000s
sys     0m0.001s

real    0m0.001s
user    0m0.001s
sys     0m0.000s

real    0m0.017s
user    0m0.001s
sys     0m0.006s

real    0m0.002s
user    0m0.002s
sys     0m0.000s
danielc@danielc-VirtualBox:~$
```

