

Universidad Autónoma de Baja California  
Facultad de Ciencias Químicas e Ingeniería



Héctor Miguel Macías Baltazar

1272124

Administración de Sistemas  
operativos

351

Taller 6: Procesos

25/10/2023

## Introducción.

Unix es un sistema multiproceso por tiempo compartido. Aunque en cualquier momento muchos procesos parecen estar ejecutándose concurrentemente desde el punto de vista del proceso, este tiene el acceso y control de todos los recursos del sistema como si fuera único. La realidad es que la mayoría de los sistemas Unix corren en plataformas que tienen una unidad de procesamiento capaz de soportar muchos procesos activos; sin embargo, en un instante dado de tiempo solamente un proceso puede estar en ejecución.

Los sistemas operativos Unix cambian rápidamente (en milisegundos) el proceso actual en ejecución, aparentando así que trabajan en forma concurrente; es decir, aparentan que trabajan en varios procesos al mismo tiempo pero en realidad, existe una distribución temporal de la asignación de CPU entre los diferentes procesos que compiten por ejecutarse. La conmutación temporal de procesos, está basada en un sistema de colas FIFO multinivel con actualización de prioridad.

Los sistemas con múltiples unidades de procesamiento, las cuales por definición pueden soportar un verdadero procesamiento concurrente se dice que tienen capacidad de multiprocesamiento. En la figura puede ver que los procesos tienen una estructura jerárquica en árbol:

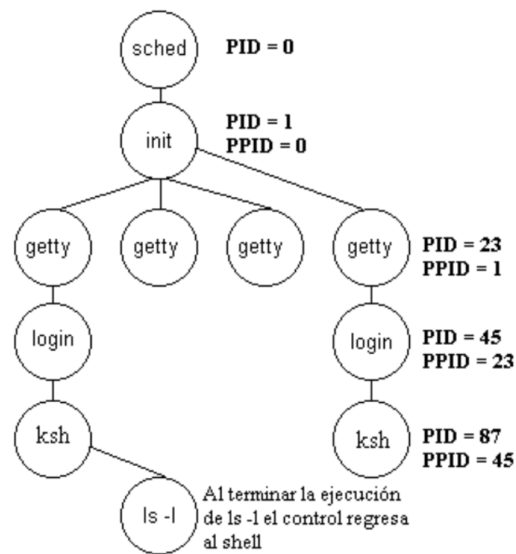


Figura 1: Ejemplo del árbol de procesos de UNIX.

Existen dos tipos de procesos:

1. Procesos del sistema: Son los procesos que actúan sin que el usuario los solicite. También se les conoce como daemons. Pueden ser de dos tipos:
  - a. Procesos permanentes o de larga duración: Se crean al arrancar el sistema y permanecen activos hasta que se termina la conexión. Su función es realizar las actividades del sistema.

- b. Procesos transitorios: Nacen y mueren cuando el sistema efectúa tareas propias, independientes de los usuarios.
2. Procesos de usuario: Son los procesos creados cuando el usuario ejecuta comandos.

Los procesos tienen los siguientes atributos:

- Cada proceso tiene un identificador único (PID). Este es un entero no negativo asignado por el sistema. Garantiza que el proceso sea único dentro del sistema.
- Un proceso tiene asignados tres segmentos de memoria: Segmento de código, Segmento de datos (datos estáticos) y Segmento de Stack (datos dinámicos).
- Identificador del usuario y grupo al que pertenece.
- Identificadores de otros procesos relacionados con él
- Datos de los sucesos que le harán despertar y señales pendientes.
- Tamaño del proceso
- Datos de su planificación.
- Terminal Original

#### *Prioridades de los Procesos.*

Una manera de minimizar los tiempos de respuesta del procesador para aprovechar de forma eficiente los recursos del sistema es que mientras un usuario realiza sus procesos otros usuarios pueden tener tareas en ejecución de manera simultánea.

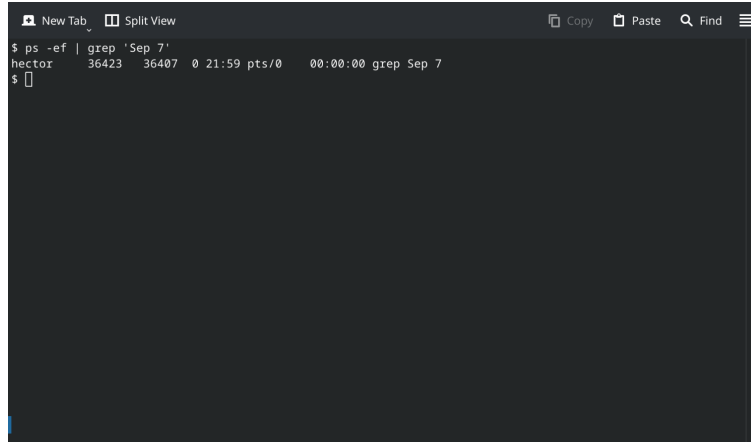
- Dejar trabajos en baja prioridad en el sistema es una buena manera de utilizar los recursos eficientemente.
- Los procesos interactivos (los que requieren entrada/salida) necesitan mayor prioridad.
- El sistema ajusta las prioridades de los procesos dinámicamente, de tal forma que se puede ajustar a cambios en los procesos.
- Los procesos que no utilizan el procesador por un periodo aumentan su prioridad.
- El comando nice permite dejar un proceso en baja prioridad. Dependiendo de la versión de UNIX acepta un rango diferente de prioridades. Pero los números más altos tienen prioridad más baja.

#### **Desarrollo.**

Tomando en cuenta la información provista por el manual de prácticas, realizamos las siguientes actividades para poner en práctica la teoría revisada:

1. Genere un listado completo de todos los procesos que están en el sistema y muestre la información completa de todos los que se empezaron a ejecutar el 7 de septiembre en una sola línea.

```
ps -ef | grep 'Sep 7'
```

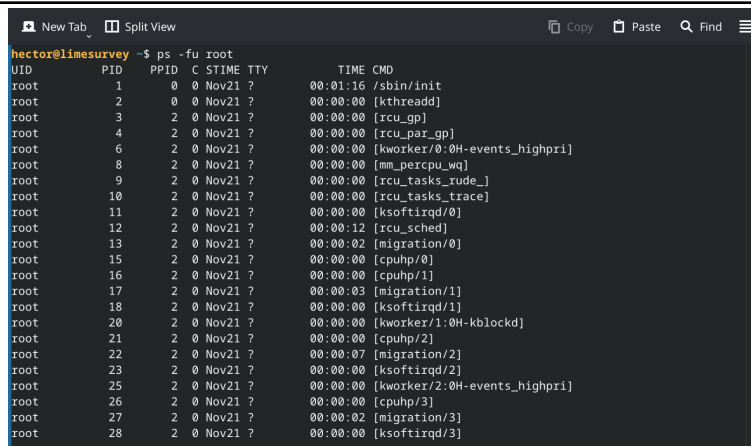


```
New Tab Split View Copy Paste Find
$ ps -ef | grep 'Sep 7'
hector 36423 36407 0 21:59 pts/0 00:00:00 grep Sep 7
$
```

Imagen 1: Listado de todos los procesos del sistema.

2. Qué están haciendo los procesos que actualmente está ejecutando maestro. (Comando).

```
ps -fu root
```

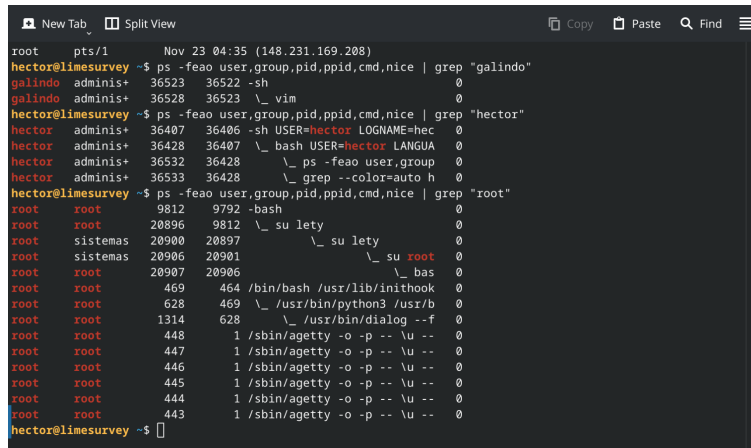


```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ ps -fu root
  UID      PID     PPID  C  STIME TTY          TIME CMD
root         1         0  0 Nov21 ?        00:01:16 /sbin/init
root         2         0  0 Nov21 ?        00:00:00 [kthreadd]
root         3         2  0 Nov21 ?        00:00:00 [rcu_gp]
root         4         2  0 Nov21 ?        00:00:00 [rcu_par_gp]
root         6         2  0 Nov21 ?        00:00:00 [kworker/0:0H-events_highpri]
root         8         2  0 Nov21 ?        00:00:00 [mm_percpu_wq]
root         9         2  0 Nov21 ?        00:00:00 [rcu_tasks_rude_]
root        10         2  0 Nov21 ?        00:00:00 [rcu_tasks_trace]
root        11         2  0 Nov21 ?        00:00:00 [ksoftirqd/0]
root        12         2  0 Nov21 ?        00:00:12 [rcu_sched]
root        13         2  0 Nov21 ?        00:00:02 [migration/0]
root        15         2  0 Nov21 ?        00:00:00 [cpuhp/0]
root        16         2  0 Nov21 ?        00:00:00 [cpuhp/1]
root        17         2  0 Nov21 ?        00:00:03 [migration/1]
root        18         2  0 Nov21 ?        00:00:00 [ksoftirqd/1]
root        20         2  0 Nov21 ?        00:00:00 [kworker/1:0H-kblockd]
root        21         2  0 Nov21 ?        00:00:00 [cpuhp/2]
root        22         2  0 Nov21 ?        00:00:07 [migration/2]
root        23         2  0 Nov21 ?        00:00:00 [ksoftirqd/2]
root        25         2  0 Nov21 ?        00:00:00 [kworker/2:0H-events_highpri]
root        26         2  0 Nov21 ?        00:00:00 [cpuhp/3]
root        27         2  0 Nov21 ?        00:00:02 [migration/3]
root        28         2  0 Nov21 ?        00:00:00 [ksoftirqd/3]
```

Imagen 2: Lista de procesos del maestro (root).

3. Genere un listado con el número de proceso, número del proceso padre, comando en ejecución y prioridad de tres de sus compañeros.

```
ps -feao user, group, pid, ppid, cmd, nice | grep "usuario"
```



```
root pts/1 Nov 23 04:35 (148.231.169.208)
hector@limesurvey ~$ ps -feao user,group,pid,ppid,cmd,nice | grep "galindo"
galindo adminis+ 36523 36522 -sh 0
galindo adminis+ 36528 36523 \_ vim 0
hector@limesurvey ~$ ps -feao user,group,pid,ppid,cmd,nice | grep "hector"
hector adminis+ 36407 36406 -sh USER=hector LOGNAME=hec 0
hector adminis+ 36428 36407 \_ bash USER=hector LANGUA 0
hector adminis+ 36532 36428 \_ ps -feao user,group 0
hector adminis+ 36533 36428 \_ grep --color=auto h 0
hector@limesurvey ~$ ps -feao user,group,pid,ppid,cmd,nice | grep "root"
root root 9812 9792 -bash 0
root root 20896 9812 \_ su lety 0
root sistemas 20900 20897 \_ su lety 0
root sistemas 20906 20901 \_ su root 0
root root 20907 20906 \_ \_ bas 0
root root 469 464 /bin/bash /usr/lib/inihook 0
root root 628 469 \_ /usr/bin/python3 /usr/b 0
root root 1314 628 \_ /usr/bin/dialog --f 0
root root 448 1 /sbin/agetty -o -p -- \u -- 0
root root 447 1 /sbin/agetty -o -p -- \u -- 0
root root 446 1 /sbin/agetty -o -p -- \u -- 0
root root 445 1 /sbin/agetty -o -p -- \u -- 0
root root 444 1 /sbin/agetty -o -p -- \u -- 0
root root 443 1 /sbin/agetty -o -p -- \u -- 0
hector@limesurvey ~$
```

Imagen 3: Procesos de dos de mis compañeros y yo.

4. Explique la diferencia entre las opciones de ps e,f,l y j.

- -e: Muestra información de todos los procesos del sistema.
- -f: Proporciona una visualización más detallada, incluyendo la relación jerárquica entre procesos (padre e hijo).
- -l: Ofrece una salida más detallada, incluyendo información sobre los recursos utilizados por cada proceso.
- -j: Muestra información sobre los grupos de procesos.

5. Explique la diferencia entre las opciones de ps a y u.

- -a: Muestra la información de todos los usuarios, no solo del usuario actual. Proporciona una vista global de los procesos en el sistema.
- -u: Muestra información detallada sobre los procesos y se centra en el usuario actual. Proporciona detalles específicos del usuario que ejecutó el comando.

6. Explique qué es lo que hace la opción de ps t y u. Si tiene dos sesiones de ssh abiertas con el mismo username.

-t.

Este parámetro lista los procesos asociados con la terminal.

```
hector@limesurvey ~$ ps -t
  PID TTY          STAT       TIME COMMAND
 36407 pts/0    Ss          0:00 -sh
 36428 pts/0    S           0:00 bash
 36539 pts/0    R+          0:00 ps -t
hector@limesurvey ~$
```

Imagen 6.1: Comando ps con -t.

-u.

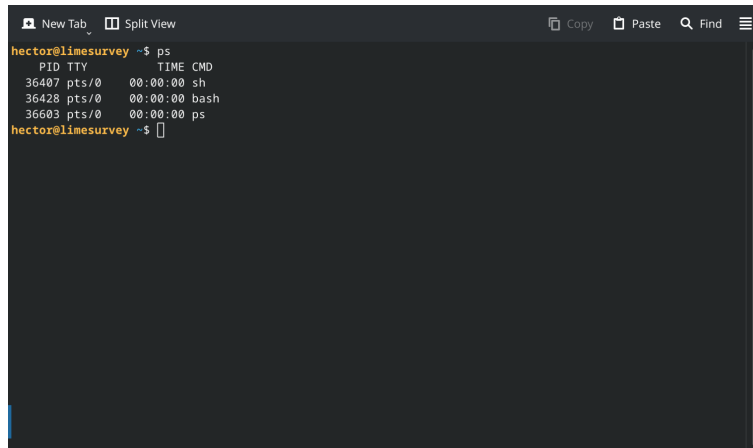
Este parámetro nos permite visualizar los procesos del usuario independientemente de la terminal que estemos usando.

```
hector@limesurvey ~$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
hector    36407  0.0  0.0   2480   1564 pts/0    Ss   21:39   0:00 -sh
hector    36428  0.0  0.0   5176   4452 pts/0    S    22:02   0:00 bash
hector    36541  0.0  0.0   6756   2960 pts/0    R+   22:18   0:00 ps -u
hector@limesurvey ~$
```

Imagen 6.2: Comando ps con -u.

7. ¿Qué procesos muestra al ejecutar ps?

Los procesos activos de mi usuario:

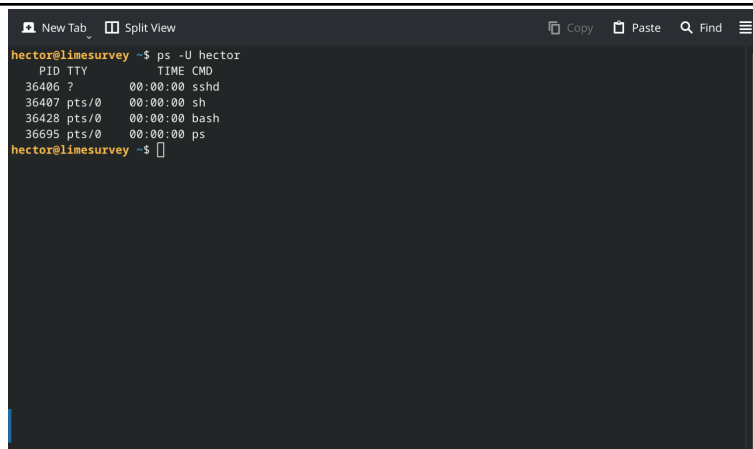


```
hector@limesurvey ~$ ps
  PID TTY          TIME CMD
 36407 pts/0    00:00:00 sh
 36428 pts/0    00:00:00 bash
 36603 pts/0    00:00:00 ps
hector@limesurvey ~$
```

Imagen 7: Comando ps.

8. ¿Qué opción de ps debería de usar para ver todos los procesos de un usuario?

```
ps -u hector
```

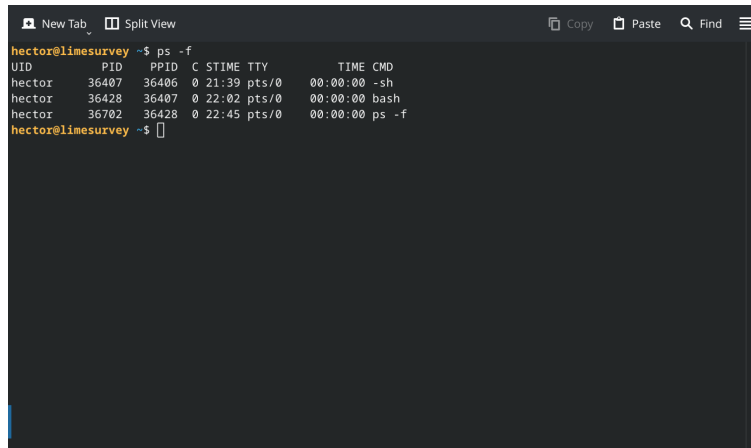


```
hector@limesurvey ~$ ps -u hector
  PID TTY          TIME CMD
 36406 ?        00:00:00 sshd
 36407 pts/0    00:00:00 sh
 36428 pts/0    00:00:00 bash
 36695 pts/0    00:00:00 ps
hector@limesurvey ~$
```

Imagen 8: Procesos de mi usuario.

9. ¿Cómo identifica a los procesos que el usuario está ejecutando en cada terminal?

Revisando la columna TTY en el comando ps -f.



```
hector@limesurvey ~$ ps -f
UID          PID    PPID  C  TIME TTY          TIME CMD
hector      36407   36406  0  21:39 pts/0    00:00:00 -sh
hector      36428   36407  0  22:02 pts/0    00:00:00 bash
hector      36702   36428  0  22:45 pts/0    00:00:00 ps -f
hector@limesurvey ~$
```

Imagen 9: En la columna TTY se muestran los procesos en pts/0.

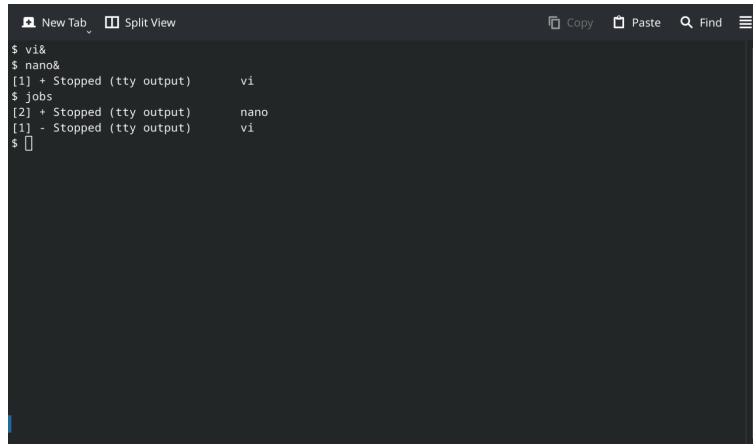
10. ¿Cuál es el significado de TODAS las columnas de formato que maneja `ps -o?` (Sólo las que no están explicadas en este material).

- `%cpu`: Porcentaje de uso de la CPU por el proceso.
- `%mem`: Porcentaje de uso de la memoria física por el proceso.
- `etime`: Tiempo total transcurrido desde que se inició el proceso. La forma exacta de representación puede variar (días-horas:minutos:segundos).
- `etimes`: Tiempo total transcurrido desde que se inició el proceso, expresado en segundos.
- `c`: La clase de la tarea. Generalmente, se muestra como "TS" (sistema de tiempo real) o "FF" (tareas de tiempo compartido).
- `stime`: Tiempo total del sistema consumido por el proceso desde su inicio.
- `utime`: Tiempo total de usuario consumido por el proceso desde su inicio.
- `vsz`: Tamaño virtual de la memoria del proceso en kilobytes.
- `nlwp`: Número de hilos de ejecución ligera (LWP, por sus siglas en inglés) asociados con el proceso.

11. Ejecute dos comandos en background (los que quiera).

```
vi&
nano&
```

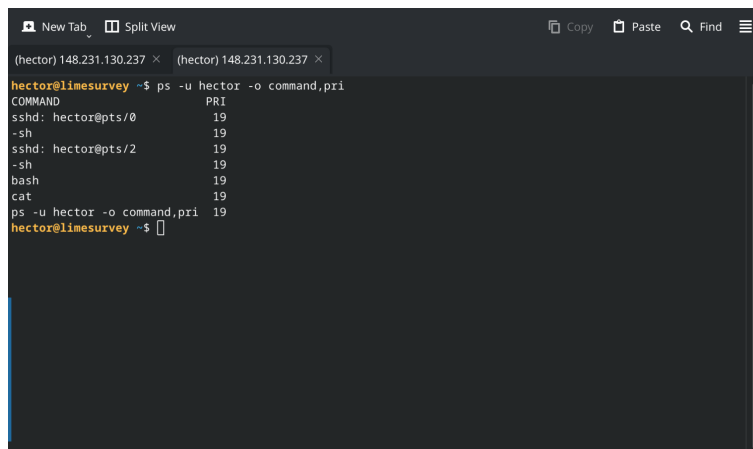


A terminal window with a dark background and light text. The terminal shows the execution of 'vi &' and 'nano &', followed by 'jobs' which lists two background processes: '[1] + Stopped (tty output) vi' and '[2] + Stopped (tty output) nano'.

```
$ vi&
$ nano&
[1] + Stopped (tty output)    vi
$ jobs
[2] + Stopped (tty output)    nano
[1] - Stopped (tty output)    vi
$
```

Imagen 11: Los procesos vi y nano fueron enviados al background.

12. Ejecute el comando `cat >lista`, ¿Qué prioridad tiene asignada?

A terminal window showing the output of the 'ps -u hector -o command,pri' command. The output lists several processes including 'sshd', 'sh', 'bash', 'cat', and 'ps', all with a priority of 19.

```
(hector) 148.231.130.237 x (hector) 148.231.130.237 x
hector@limesurvey ~$ ps -u hector -o command,pri
COMMAND      PRI
sshd: hector@pts/0      19
sh            19
sshd: hector@pts/2      19
sh            19
bash          19
cat           19
ps -u hector -o command,pri 19
hector@limesurvey ~$
```

Imagen 12: cat tiene una prioridad de 19.

13. Mate el proceso anterior.

```
kill -9 38518
```

```
New Tab Split View Copy Paste Find
(hector) 148.231.130.237 x (hector) 148.231.130.237 x
hector@limesurvey ~$ ps -u hector
  PID TTY          TIME CMD
 38445 ?        00:00:00 sshd
 38446 pts/0    00:00:00 sh
 38460 pts/0    00:00:00 vi
 38461 pts/0    00:00:00 nano
 38518 pts/0    00:00:00 cat
 38562 pts/0    00:00:00 bash
 38594 ?        00:00:00 sshd
 38595 pts/2    00:00:00 sh
 38601 pts/2    00:00:00 bash
 38642 pts/2    00:00:00 ps
hector@limesurvey ~$ kill -9 38518
hector@limesurvey ~$ ps -u hector
  PID TTY          TIME CMD
 38445 ?        00:00:00 sshd
 38446 pts/0    00:00:00 sh
 38460 pts/0    00:00:00 vi
 38461 pts/0    00:00:00 nano
 38562 pts/0    00:00:00 bash
 38594 ?        00:00:00 sshd
 38595 pts/2    00:00:00 sh
 38601 pts/2    00:00:00 bash
```

Imagen 13: Eliminación del proceso cat.

14. Vuelva a ejecutar cat>lista pero con menor prioridad.

```
nice -n 20 cat>lista
```

```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ nice -n 20 cat>lista
^Z
[1]+  Stopped                  nice -n 20 cat > lista
hector@limesurvey ~$ ps -u hector -o command,pri,ni
COMMAND                                PRI  NI
sshd: hector@pts/0                      19   0
-sh                                     19   0
vi                                       19   0
nano                                    19   0
bash                                    19   0
cat                                     0   19
ps -u hector -o command,pri             19   0
hector@limesurvey ~$
```

Imagen 14: Proceso cat con prioridad 20.

15. ¿Qué prioridad le fue asignada?

Una prioridad de 0.

16. Una vez más ejecute cat>lista, pero ahora en el background .

```
cat>lista&
```

```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ cat>lista&
[2] 38651
hector@limesurvey ~$ ps -u hector -o command,pri
COMMAND                                PRI
sshd: hector@pts/0                      19
sh                                       19
vi                                       19
nano                                    19
bash                                    19
cat                                     0
cat                                     19
ps -u hector -o command,pri            19

[2]+  Stopped                  cat > lista
hector@limesurvey ~$
```

Imagen 16: Comando cat en el background.

17. ¿Cuál es su prioridad ahora?

Tiene una prioridad de 19.

18. Verifique que el comando en background esté en la lista de procesos.

```
ps -l
```

```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ ps -l
F S  UID      PID  PPID  C PRI  NI ADDR SZ  WCHAN  TTY          TIME CMD
0 S  1039    38446   38445  0  80   0 -  620 -  pts/0    00:00:00 sh
0 T  1039    38460   38446  0  80   0 - 1234 -  pts/0    00:00:00 vi
0 T  1039    38461   38446  0  80   0 -  819 -  pts/0    00:00:00 nano
0 S  1039    38562   38446  0  80   0 - 1294 -  pts/0    00:00:00 bash
0 T  1039    38648   38562  0  99  19 -  632 -  pts/0    00:00:00 cat
0 T  1039    38651   38562  0  80   0 -  632 -  pts/0    00:00:00 cat
0 R  1039    38656   38562  0  80   0 - 1669 -  pts/0    00:00:00 ps
hector@limesurvey ~$
```

Imagen 18: Usando el comando ps -l verificamos que cat está en la lista.

19. Verifique que el comando en background esté en la lista de tareas (jobs).

```
jobs
```

```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ jobs
[1]-  Stopped                  nice -n 20 cat > lista
[2]+  Stopped                  cat > lista
hector@limesurvey ~$
```

Imagen 19: Verificamos que cat está en jobs.

20. Pase una de las tareas al foreground (use el número de tarea).

```
fg %1
```

```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ jobs
[1]-  Stopped                  nice -n 20 cat > lista
[2]+  Stopped                  cat > lista
hector@limesurvey ~$ fg %1
nice -n 20 cat > lista

```

Imagen 20: Pasamos la tarea 1 al foreground.

21. Pase la otra tarea al foreground, pero ahora use el número de PID.

No se puede usar el comando fg con el número PID de un proceso.

22. Envíe otro comando al background.

```
nano&
```

```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ nano&
[3] 38705
hector@limesurvey ~$
```

Imagen 22: Enviamos el proceso nano al background.

23. Finalice este proceso.

```
kill -9 38705
```

```
New Tab Split View Copy Paste Find
hector@limesurvey ~$ ps -u hector
  PID TTY          TIME CMD
 38445 ?        00:00:00 sshd
 38446 pts/0    00:00:00 sh
 38562 pts/0    00:00:00 bash
 38648 pts/0    00:00:00 cat
 38651 pts/0    00:00:00 cat
 38705 pts/0    00:00:00 nano
 38708 pts/0    00:00:00 ps
hector@limesurvey ~$ kill -9 38705
hector@limesurvey ~$ ps -u hector
  PID TTY          TIME CMD
 38445 ?        00:00:00 sshd
 38446 pts/0    00:00:00 sh
 38562 pts/0    00:00:00 bash
 38648 pts/0    00:00:00 cat
 38651 pts/0    00:00:00 cat
 38709 pts/0    00:00:00 ps
[3]+  Killed                  nano
hector@limesurvey ~$
```

Imagen 23: Terminamos el proceso nano con kill.

## Conclusiones.

En esta práctica aprendí que los procesos en UNIX los podemos reconocer usando identificadores como sus PID o sus PPID. Aprendí que podemos enviar procesos al background y foreground a nuestra disposición, así como configurar la prioridad de los procesos y terminarlos súbitamente.

## Referencias.

Corbalán, A. S. (2022, 8 abril). Procesos en Linux. Información y Administración. *Antonio Sánchez Corbalán*.

<https://sanchezcorbalan.es/procesos-en-linux-informacion-y-administracion/>

*El control de procesos en UNIX*. (s. f.).

<http://www.hpca.ual.es/~vruiz/docencia/cursos/linux/html/node47.html>

González, A. J. (s. f.). *Control de procesos en UNIX*.

<http://profesores.elo.utfsm.cl/~agv/elo330/2s12/lectures/ControlProcesos.html>

Núñez, E. A. (2023, 13 abril). 20 Comandos para administrar y gestionar fácilmente procesos en Linux. *OpenWebinars.net*.

<https://openwebinars.net/blog/20-comandos-para-administrar-y-gestionar--facilmente-los-procesos-linux/>