**Gestión y Manipulación de Procesos**

**¿Qué es un proceso?**

**Un proceso es un programa en ejecución. Mientras que un programa es un archivo con instrucciones guardado en el disco, un proceso es la instancia de ese programa que ha sido cargada en memoria y está siendo ejecutada por el CPU.**

**Por ejemplo, cuando ejecutás firefox, el programa que está en /usr/bin/firefox se convierte en un proceso que ocupa memoria, accede a recursos y realiza tareas.**

**Diferencias entre programa y proceso**

* **Programa: archivo estático, sin ejecución, guardado en el sistema de archivos.**
* **Proceso: ejecución activa del programa, con uso de recursos como CPU, RAM, archivos abiertos, etc.**

**Características de un proceso**

**Cada proceso tiene:**

* **Un PID (Process ID): número único que lo identifica.**
* **Un proceso padre: el proceso que lo creó. Tiene PPID**
* **Su propio espacio de memoria.**
* **Recursos asociados (archivos, terminal, etc.).**
* **Un estado (activo, dormido, detenido, etc.).**

**Los procesos pueden crear procesos hijos, y esta jerarquía puede visualizarse como un árbol.**

**Estados de los procesos**

* **Running (R): en ejecución o listo para ejecutarse.**
* **Sleeping (S): esperando algún evento (como entrada/salida).**
* **Stopped (T): detenido por una señal o por el usuario.**
* **Zombie (Z): terminó pero su padre aún no leyó su estado de salida.**

**Señales en Linux**

Una **señal** es una notificación enviada a un proceso para indicarle que ocurra una acción. Las más comunes son:

|  |  |  |
| --- | --- | --- |
| Señal | Nombre | Acción |
| 1 | SIGHUP | Reiniciar proceso |
| 2 | SIGINT | Interrupción |
| 9 | SIGKILL | Forzar terminación inmediata |
| 15 | SIGTERM | Terminar proceso (por defecto) |
| 18/19 | SIGCONT/SIGSTOP | Reanudar o detener proceso |
|  |  |  |
|  |  |  |

Ejemplo: kill -SIGTERM 1234 o kill -15 1234

Kill se usa para enviar una señal a un proceso

**Prioridad y planificación de procesos**

Linux usa un sistema de prioridades para decidir qué proceso se ejecuta primero.

* La **prioridad base** se define con el valor **nice**, de -20 (mayor prioridad) a 19 (menor prioridad).
* Sólo el **root** puede ejecutar procesos con valores negativos.

Ejemplos:

* nice -n 10 ./mi\_script.sh

Actividad: Investigacion de comandos sobre procesos.

Buscar comandos que ayuden a realizar las siguientes tareas:  
-**Comandos para gestión de procesos**

-**Para buscar procesos**

-**Para finalizar procesos**

-**Para suspender o reanudar**

-**Para cambiar prioridad**

Actividad 2: Responder las siguientes preguntas

¿Cómo se identifica el PID de un proceso en ejecución?

¿Qué comando permite monitorear en tiempo real los procesos?

¿Qué hacen los comandos nice y renice?

¿Qué pasa si un proceso queda "zombie"?

¿Qué diferencia hay entre un proceso en foreground y background?

¿Cómo ver los procesos de un usuario específico?

¿Qué pasa si matás un proceso crítico del sistema?

Respuestas

 **Gestión general de procesos:**

* ps: Muestra información sobre los procesos activos.
* top: Monitorea en tiempo real el uso del CPU/memoria por procesos.
* htop: Versión mejorada de top (requiere instalación).
* jobs: Lista los procesos en segundo plano del shell actual.
* nice, renice: Gestionan prioridades.

 **Buscar procesos:**

* ps aux | grep nombre\_proceso: Busca un proceso por nombre.
* pgrep nombre: Muestra el PID de procesos que coinciden con el nombre.
* pidof nombre: Devuelve el PID de un programa.

 **Finalizar procesos:**

* kill PID: Envía una señal (por defecto SIGTERM) para terminar un proceso.
* kill -9 PID: Fuerza la terminación inmediata del proceso.
* pkill nombre: Mata todos los procesos que coincidan con el nombre.
* killall nombre: Igual a pkill, pero también puede finalizar procesos con nombres exactos.

 **Suspender o reanudar procesos:**

* kill -STOP PID: Suspende un proceso.
* kill -CONT PID: Reanuda un proceso suspendido.
* Ctrl + Z: Suspende un proceso en primer plano (y lo pone en segundo plano).
* bg: Reanuda el proceso en segundo plano.
* fg: Trae el proceso al primer plano.

 **Cambiar la prioridad de un proceso:**

* nice -n N comando: Inicia un proceso con prioridad N (de -20 a 19).
* renice -n N -p PID: Cambia la prioridad de un proceso ya iniciado.

Actividad 2:

 **¿Cómo se identifica el PID de un proceso en ejecución?**  
Usando comandos como ps, pidof, pgrep, o top. Cada proceso en ejecución tiene un número identificador único llamado PID.

 **¿Qué comando permite monitorear en tiempo real los procesos?**

* top: Muestra una vista dinámica de los procesos.
* htop: Similar, pero con una interfaz más amigable (requiere instalación).

 **¿Qué hacen los comandos nice y renice?**

* nice: Ejecuta un proceso con una prioridad definida.
* renice: Cambia la prioridad de un proceso que ya está en ejecución.  
  Cuanto más bajo es el valor (mínimo -20), mayor es la prioridad del proceso.

 **¿Qué pasa si un proceso queda "zombie"?**  
Un proceso "zombie" ha terminado, pero su información de salida no fue recogida por su proceso padre. Ocupa una entrada en la tabla de procesos, pero no consume recursos. Si hay muchos, puede saturar el sistema.

 **¿Qué diferencia hay entre un proceso en foreground y background?**

* **Foreground (primer plano):** Bloquea el uso del terminal hasta que finaliza.
* **Background (segundo plano):** Permite seguir usando la terminal mientras el proceso corre. Se ejecutan con comando &.

 **¿Cómo ver los procesos de un usuario específico?**

* ps -u usuario
* top -u usuario
* pgrep -u usuario

 **¿Qué pasa si matás un proceso crítico del sistema?**  
Puede causar inestabilidad, pérdida de datos, cierre inesperado del sistema o incluso que el sistema deje de funcionar correctamente y necesite reiniciarse. Ejemplo: matar el proceso init o systemd puede reiniciar o apagar el sistema.