Ejercicios Comandos Gestión procesos en Linux -resuelto

# Comando ps

1. Muestra por terminal los procesos activos en el sistema.

**Ps aux**

1. Inicia sesión con el usuario User02. Averigua cuales son los procesos activos de root.

**sudo su - User02**

**ps aux | grep root**

**exit**

1. Vuelve a la sesión de alumno. Mostrar, por pantalla, solo el listado de los procesos o comandos, el pid y el consumo de CPU de cada uno de ellos.

**ps -eo cmd,pid,%cpu**

1. Idem que el ejercicio anterior pero ordenado por consumo de cpu.

**ps -eo cmd,pid,%cpu --sort=-%cpu**

1. Idem que el ejercicio anterior pero solo los 3 procesos que más cpu consumen.

**ps -eo cmd,pid,%cpu --sort=-%cpu | head -n 4**

1. Mostrar, por pantalla, solo el pid , el comando y la cantidad de memoria ram que consumen los procesos.

**ps -eo pid,cmd,%mem**

1. Idem que el ejercicio anterior pero ordenador por cantidad de memoria consumida de mayor a menor

**ps -eo pid,cmd,%mem --sort=-%mem**

1. Idem que el anterior pero solo los 5 procesos que más cantidad de memoria consumen.

**ps -eo pid,cmd,%mem --sort=-%mem | head -n 6**

1. Mostrar el listado de todos los procesos que se están ejecutando incluyendo su jerarquía, es decir, de quien dependen.

**pstree**

1. Mostrar los procesos de systemd incluyendo la jerarquía.

**pstree -p | grep systemd**

1. Mostrar por pantalla el identificador (pid) , el nombre (comm) y la prioridad (ni) de todos los procesos del sistema.

**ps -eo pid,comm,ni**

1. Idem pero ordenando por máxima prioridad los procesos.

**ps -eo pid,comm,ni --sort=-ni**

1. Idem pero mostrando los 5 procesos con máxima prioridad.

**ps -eo pid,comm,ni --sort=-ni | head -n 6**

1. Mostrar el pid, cantidad de cpu , cantidad de memoria, usuario y comando de todos los procesos secundarios de un proceso systemd.

**pgrep -P $(pgrep -f systemd) -a | awk '{print $1, $(NF-4), $(NF-3), $(NF-2), $(NF)}'**

# Comando pstree

1. Mostrar el árbol de procesos activos en el sistema.

**pstree**

1. Mostrar el árbol de procesos activos del sistema deshabilitando los nombres repetidos.

**pstree -U**

1. Mostrar el árbol de procesos incluyendo también los argumentos de la línea de comandos en la salida.

**pstree -a**

1. Mostrar resaltado el proceso cuyo pid darás como opción.

**pstree -p <PID>**

1. Mostrar los ID de los grupos de procesos en la salida.

**pstree -g**

1. Ordenar los procesos en función de los PID

**pstree -p --sort=p**

1. Mostrar el árbol de procesos del usuario alumno

**pstree -u alumno**

1. mostrar sólo la información de los padres e hijos del proceso 1701.

**pstree -p 1701**

# Comando jobs

1. ¿Cuándo se debe lanzar un proceso en segundo plano o background?

**Se debe lanzar un proceso en segundo plano cuando se desea ejecutar un comando sin bloquear la terminal y permitir que otros comandos se ejecuten mientras el proceso en segundo plano continúa.**

1. ¿Cuándo se lanzara un proceso en primer plano o foreground?

**Se lanzará un proceso en primer plano cuando se desea ejecutar un comando y bloquear la terminal hasta que ese proceso termine.**

1. Ejecuta los siguientes pasos en el terminal de la maquina virtual:
   1. Ejecuta el comando Firefox

**firefox**

* 1. Pulsa la combinación de teclas Ctrl+Z. ¿Qué ocurre?

**Firefox se detiene y vuelve al shell.**

* 1. Ejecuta el comando jobs. ¿Qué muestra?

**Muestra el trabajo suspendido, en este caso, el proceso de Firefox.**

* 1. Ejecuta el comando gparted como root.

**sudo gparted**

* 1. Pulsa la combinación de teclas Ctrl+Z.

**Gparted se detiene y vuelve al shell.**

* 1. Accede a la pantalla donde esta Firefox iniciado y, ¿Qué ocurre?

**No se puede acceder a la pantalla de Firefox, ya que está suspendido.**

* 1. Vuelve al terminal y ejecuta el comando jobs. ¿Qué observas?

**Muestra dos trabajos suspendidos, uno para Firefox y otro para gparted.**

* 1. Observa que la línea donde aparece Firefox tiene un número. Ejecuta la orden fg numero\_linea\_firefox. ¿Qué ha ocurrido?

**fg 1**

**Firefox vuelve al primer plano.**

* 1. Pulsa la combinación de teclas Ctrl+Z.

**Firefox se detiene y vuelve al shell.**

* 1. Ejecuta el comando jobs y observa que aparece.

**Muestra el trabajo suspendido de Firefox.**

* 1. Ejecuta el comando killall con el parámetro -9 y el nombre del proceso firefox.

¿Qué ocurre?

**Firefox se cierra al momento**

* 1. Ejecuta el comando jobs. ¿Qué muestra?

**No muestra nada, ya que no hay trabajos en segundo plano.**

* 1. Ejecuta el comando killall con el parámetro -9 y el nombre del proceso gparted.

**sudo killall -9 gparted**

* 1. Ejecuta el comando jobs.
  2. Ejecuta la siguiente orden: yes > /dev/null. ¿Qué ocurre?

**El comando yes se ejecuta en segundo plano y genera una salida infinita.**

* 1. Pulsa la combinación de teclas Ctrl+Z y ejecuta el comando jobs.

**Muestra el trabajo en segundo plano de yes.**

* 1. Ejecuta la orden : killall -9 yes

**Detiene el proceso yes.**

* 1. Ejecuta la siguiente orden: yes > /dev/null&. ¿Qué ocurre ahora? Ejecuta el comando jobs.

**El comando yes se ejecuta en segundo plano.**

* 1. Ejecuta el comando: fg 1. ¿Qué observas? Pulsa la combinación de teclas Ctrl+Z y ejecuta el comando jobs.

**El comando yes vuelve al primer plano. Al pulsar Ctrl+Z, se detiene y vuelve al shell.**

* 1. Ejecuta el comando: bg 1. Ejecuta el comando jobs.

**El comando yes vuelve a ejecutarse en segundo plano.**

* 1. Ejecuta el comando ps. Observa la línea donde aparece el comando yes. ¿Qué indica?

**Muestra información sobre el proceso yes, incluyendo su PID.**

* 1. Ejecuta el comando kill con la opción -9 y el numero PID que aparece en la línea de yes.

**Detiene el proceso yes en el momento**