

## PROCESOS Y COMPRIMIR

### Envío a segundo plano y eliminación de procesos:

El comando `yes` envía una serie de 'y' a la salida estándar, `/dev/null` es un dispositivo 'agujero negro' todo lo que se le pase 'desaparecerá', así `'yes > /dev/null'` no mostrara nada en la salida estándar ya que todo se va a `/dev/null`. Para enviar un proceso a segundo plano lo llamamos desde la línea de comandos seguido de '&'.

1. Ejecuta el comando `yes > /dev/nul` y ejecuta `ctrl +c` para parar el proceso
2. Vuelve a ejecutar `yes > /dev/null &`. ¿Qué es el número que va entre corchetes [ n°] y el número que lo sigue?
3. A continuación, ejecuta el comando `Jobs`. ¿En qué estado está el proceso?
4. Ahora ejecuta el comando `ps`. ¿Cuánto tiempo lleva el proceso ejecutándose?
5. Vamos a matar el proceso (terminar) con el comando `kill`. Procede a matar el proceso puedes utilizar el número de tarea o si identificador de proceso. Elige una y terminar el proceso `yes`.
6. Ejecuta `jobs` para ver que nos dice sobre el proceso.

### Parada y relanzamiento de tareas:

1. Para poner una tarea en segundo plano después de haberla lanzada usamos `ctrl +z`. Ejecuta `yes > /dev/null` y a continuación pulsa `CTRL + Z`.
2. Vamos a traer a primer plano el proceso `yes` utilizando el comando `fg` y volvemos a pulsar `CTRL + Z`
3. Ahora lo mandamos a segundo plano utilizando el comando `bg` y comprobar con `jobs` en qué estado se encuentra.
4. Lo volvemos a traer a primer plano y lo volvemos a parar con `CTRL +Z`.
5. Mata el proceso utilizando el número de tarea.
6. Vamos a lanzar el proceso `yes > /dev/null &` dos veces, esto ejecutará el proceso en segundo plano dos veces. Y de nuevo lanzamos el proceso `yes > /dev/null`, lanzamos este proceso en primer plano y después de lanzarlo lo paramos con `CTRL + Z` sólo este último proceso.
7. Ejecutamos el comando `jobs` y vemos en qué estado están los procesos lanzados.
8. Ejecutamos el comando `bg` que proceso lanza a segundo plano.
9. Ahora vamos a pulsar `%n°` esto trae a primer plano el proceso indicado en el número. Trae al primer plano el proceso con número de tarea 2. Pulsa `CTRL + Z`. Y ejecuta el comando `jobs` para estado de los procesos.
10. Mata el proceso con número tarea 2. Ejecuta `jobs` y comprueba los estados de los procesos.
11. A continuación, elimina todos los procesos que todavía están lanzados.

## Prioridad de ejecución:

1. El comando `nice` indica la prioridad del proceso a la hora de su creación puede ir de -20 (prioridad más baja) a 19 (prioridad más alta). Valor por defecto es cero. Ejecuta `nice` para ver qué prioridad tiene el proceso.
2. Para cambiar prioridad de un proceso ejecutamos `renice`. Vamos a abrir el editor de texto `gedit`.
3. Paramos el editor pulsando CTRL + Z. Y ejecutamos `ps -l`. ¿Qué prioridad tiene `gedit`?
4. Vamos a cambiar su prioridad con `renice`. Ejecutando `renice 1 pid` (número identificador del proceso). Vuelve a ejecutar `ps -l`.
5. Ahora vamos a ejecutar `nice -n 2 gedit`. Ejecuta de nuevo `ps -l` y indica qué prioridad tiene el editor abierto ahora.
6. Otra forma de ver procesos es con el comando `pstree`. Ejecuta dicho comando y localiza los procesos `gedit` que tenemos lanzados.
7. Ahora vamos a ejecutar el comando `top`, muestra también los procesos en tiempo real y el tiempo de CPU y memoria que están gastando.
8. Lanza el proceso `yes` en segundo plano como ya se ha hecho en la práctica y ejecuta el comando `jobs`. Mira los procesos que están en ejecución y parados.
9. Ejecuta el comando `top -U alumno` solo saldrán los procesos de usuarios. Sin cerrar `top` escribe `u` y cambia al usuario `root`. Vuelve a cambiar al usuario `alumno`. Y ahora pulsa `k` e indica el PID del proceso `yes` para que sea matado. Para salir de `top` pulsa `q`.
10. Ejecuta `jobs` y mata todos los procesos que están todavía abiertos.

## Manejo de procesos:

1. Determinar el UID efectivo de la sesión actual. Determinar el PID (Process ID) del proceso `init`.
2. Ejecutar en background (segundo plano) el comando `ping 8.8.8.8 > /dev/null`
3. Ejecutar el comando `top`, una vez iniciado enviarlo a segundo plano.
4. Mostrar todos los procesos que se están ejecutando con el mismo UID efectivo que la sesión actual
5. Mostrar un árbol sólo de los procesos que se están ejecutando con el mismo UID efectivo que la sesión actual
6. Mostrar todos los procesos ejecutando como usuario "root" (ID efectivo y real) en formato de usuario
7. Determinar el PID (Process ID) del proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`

8. Bajar al mínimo la prioridad de ejecución del proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`
9. Matar el proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`
10. Determinar qué proceso está consumiendo más CPU
11. Determinar qué proceso está consumiendo más memoria RAM
12. Determinar cuanta memoria RAM disponible (libre) posee el sistema
13. Traer a primer plano el proceso ejecutando el comando `top`

## Comando tar , gzip, filtros, direccionar y df:

1. Crear un directorio llamado "docA" que contenga los archivos "doc1.txt", "doc3.txt" y "doc77.txt". Crear un directorio llamado "docB" que contenga los archivos "doc1.txt", "doc4.txt" y "doc89.txt".
2. Listar, utilizando una única línea de comandos, los nombres de archivo que se encuentran en el directorio "docA" pero no en el directorio "docB"
3. Listar, utilizando una única línea de comandos, los nombres de archivo que se encuentran tanto en el directorio "docA" como en el directorio "docB"
4. Comprimir los directorios "docA" y "docB" utilizando el comando "gzip" en un único archivo llamado "docab.tar.gz".
5. Volcar el contenido del archivo /etc/fstab exceptuando las líneas que contengan el carácter '#'.
6. Volcar el contenido del archivo /etc/fstab exceptuando las 3 primeras líneas.
7. Buscar recursivamente dentro del directorio /etc todos los archivos cuyo nombre comienza con "net".
8. Determinar cuánto espacio en disco ocupa el directorio /home expresados en KB, MB o GB dependerá del tamaño que tengamos.