

TEMA 5

IMPORTANTE

- Si tienes dudas con cualquier comando puedes ejecutar antes: `comando --help` Ej: `chown --help`
- Intenta usar rutas absolutas, si no te dicen lo contrario, para que no pierdas el hilo de donde estas y a donde quieres ir. Ej: `nano /home/usuario/fichero`
- Rutas absolutas: tienes que escribir la ruta completa desde raíz (/)
- Rutas relativas: escribes la ruta desde el directorio donde estás
- `~`: indica el home del usuario, es lo mismo que poner `/home/nombreusuario`
- `./`: el directorio en el que estás.
- `PWD`: comando para comprobar en que directorio estoy
- La ruta **`/dev/sda/`** siempre es la del disco donde está el sistema. NO TOCAR NADA

Gestión de discos duros

- Path de los discos duros
 - SATA: `/dev/sda`
 - NVME: `/dev/nvme1`
- Path de las particiones
 - SATA: `/dev/sda1`
 - NVME: `/dev/nvme0p1`
- Tablas de particiones:
 - MBR: bastante obsoleta
 - GPT: es la que se usa ahora, no tiene limitaciones
- Pasos a la hora de montar y particionar un disco
 - Introducir el disco
 - Establecer la tabla de particiones GPT: `fdisk /dev/sdb`
 - Particionar el disco Estos pasos los podemos realizar con la herramienta **gparted** o bien de forma independiente.

De forma independiente:

- Crear tabla de particiones MBR: `fdisk /dev/sdb`
- Crear tabla de particiones GPT: `gdisk /dev/sdb`
- Particionar el disco:
 - Archivos EXT4: `mkfs.ext4 /dev/sdb1`
 - Archivo NTFS: `mkfs.ntfs /dev/sdb1`

Montaje y desmontaje manual de una partición

- Montaje: `mount -t tipodearchivo pathorigen pathdestino` Ej: `mount -t ext4 /dev/sdb1 /datos`
- Desmontaje: `umount pathdestino` Ej: `umount /datos`

Montaje automático de una partición ¡RECUERDA! cuando montas una partición de forma manual, al reiniciar el servidor deberás volverla a montar

1. Editar fichero `/etc/fstab` Ej: `sudo nano /etc/fstab`
2. Añadir línea al final del fichero `pathorigen pathdestino tipodearchivo defaults 0 0` Ej:
`/dev/sdb1 /datos ext4 defaults 0 0`

Almacenamiento RAID

RAID 0

El único que no duplica la información Mín 2 discos

- Ventajas:
 - Mayor velocidad
 - No se desperdicia cap. de almacenamiento
- Desventajas:
 - Si falla un disco se pierde todo

RAID 1 (MIRRORING)

Creación de discos espejo, almacena los datos por duplicado Mín 2 discos

- Ventajas:
 - Si falla un disco la info está en el otro disco
 - Mejora de velocidad
- Desventajas:
 - Se pierde la mitad de cap. de almacenamiento

RAID 3 y RAID 4

Se almacena por bloques de paridad. Raid 3 genera paridad bit a bit y RAID 4 genera paridad por bloques
Mín 3 discos

- Ventajas:
 - Si falla un disco la info se puede recuperar después del proceso de cálculo
 - Mejora de velocidad a nivel teórico
- Desventajas:
 - Se pierde un porcentaje de cap. de almacenamiento

RAID 5

Se almacena por bloques de paridad rotatorios. Similar a RAID 4 Mín 3 discos

- Ventajas:
 - Si falla un disco la info se puede recuperar después del proceso de cálculo
 - Mejora de velocidad respecto a RAID 3 y RAID 4
- Desventajas:
 - Se pierde un porcentaje de cap. de almacenamiento

RAID 6

Ofrece un esquema de paridad distribuido. Similar a RAID 5 Mín 3 discos

- Ventajas:
 - Mayor tolerancia a fallos, se pueden reponer hasta dos discos duros a la vez
 - Dispone de discos 'hot-spare', discos vacíos pero esperando a entrar si uno falla.
 - Mejora de velocidad respecto a RAID 3 y RAID 4
- Desventajas:
 - Se pierde cap. de almacenamiento

RAID 0+1

El más importante, produce división y duplicación de datos Mín 4 discos

- Ventajas:
 - Máximo rendimiento y protección
- Desventajas:
 - Coste elevado
 - Se pierde cap. de almacenamiento

RAID 1+0 (RAID 10)

Invertido a RAID 0+1, produce duplicación y división de datos. Elegido para bbdd de altas prestaciones Mín 4 discos

- Ventajas:
 - Mayor velocidad de escritura
- Desventajas:
 - Coste elevado
 - Se pierde cap. de almacenamiento

Crear RAID

Programa necesario: `mdadm`

Crear RAID

Parámetros:

- C: crear enlace
- l: nivel del raid
- n: número de discos `mdadm -C /dev/md/nombraid -l nivelraid -n nºdiscos listadepathsdediscos` Ej: `mdadm -C /dev/md/raidcillo0 -l 0 -n 2 /dev/sdb /dev/sdc`

Ver información del raid creado `sudo mdadm -D /dev/md/raidcillo0`

Automontar volumen RAID `/dev/md/raidcillo0 /datos ext4 defaults 0 0`

Enlaces simbólicos y duros

IMPORTANTE

- Enlace simbólico: similar a los accesos directos de Windows. Apuntan a otro enlace duro
- Enlace duro: apuntan a un fichero en particular del disco

- No se pueden crear enlaces duros de discos extraíbles o USB

Crear enlace duro

`ln fichero pathdestino` Ej: `ln documento.txt /home/usuario/enlace_duro_doc`

Enlaces duros recursivos (de todo el directorio) `cp -rl pathorigen pathdestino` Ej: `cp -rl /home/usuario/Pictures /home/usuario/Desktop/enlaceAfotos`

Crear enlace simbólico

`ln -s fichero pathdestino` Ej: `ln -s documento.txt /home/usuario/enlace_soft_doc`

Enlaces simbólicos recursivos (de todo el directorio) `cp -rs pathorigen pathdestino` Ej: `cp -rs /home/usuario/Pictures /home/usuario/Desktop/enlaceAfotos`

Eliminar enlaces

`unlink pathenlace` Ej: `unlink /home/usuario/enlace_duro_doc`

Copias de seguridad

Se usa el comando TAR ya que es recomendable comprimir la copia de seguridad para que no ocupe mucho espacio innecesario `tar -jcvf backup.tar.gz pathdirectorio` Ej: `tar -jcvf backuHome.tar.gz /home/usuario` **Extraer copia de seguridad** `tar -jxvf backup.tar.gz`

Backups incrementales

Solo agrega lo que se haya modificado `tar -cvzf backup.tgz -g snapshot.snar pathacopiar` Ej: `tar -cvzf backup/bkp1.tgz -g backup/snapshot.snar data/`

RESTAURACIÓN `tar -xvGf backup.tgz`

Programación de tareas (Crontab)

IMPORTANTE Cron comprueba los ficheros `/etc/crontab` o `/var/spool/cron`

Agregar tareas

Asistente para crear tareas: `crontab -e` Estructura: `diadelasemana mes diadelmes hora minuto comando a realizar` Ej: `* * * * * script.sh` Ej2: `0 4 * * 1 tar -zcf /backups/seguridadHome.tgz /home/`

Comandos útiles con CRON

- Editar el archivo existente: `crontab -e`
- Reemplazar fichero: `crontab fichero`
- Listar todas las tareas: `crontab -l`
- Borrar crontab configurado: `crontab -d`
- Manejar crontab de otro usuario: `crontab -u usuario`

Gestión de procesos

PS

Información sobre los procesos que se están ejecutando **Información importante:**

- PID (primera col): identificador único de cada proceso
- Terminal (segunda col): terminal en la que se está ejecutando
- Tiempo total (tercera col)
- Nombre proceso (cuarta col)

PARÁMETROS

- e: listado de los procesos
- f: listado extendido donde se ve el ppid y la hora
- ef: listado extendido
- u: procesos lanzados por un usuario
- a: todos los procesos

El más común: `ps aux`

PS TREE

Visualizar los procesos en forma de árbol `pstree`

TOP

Parecido a PS pero la información que muestra se va actualizando. Muestra el PID, el usuario, la prioridad, el nombre del proceso, el tiempo en ejecución, etc `top`

Procesos en primer y segundo plano

¿Cómo poner un proceso a correr en segundo plano? Solo es necesario añadir `&` después del comando. Ej: `sleep 10 &`

Pasar procesos de segundo plano a primer plano Cuando iniciamos un proceso nos aparece un indicador entre corchetes, esto indica el numero de proceso 1, 2, 3... según los que hayamos mandado, este numero es necesario para cambiar de plano. Ej: El resultado de este comando `root@nuria-msi:~# sleep 100 &` es: `[1] 47` Para pasar el proceso a primer plano solo necesitamos: `fg 1` y la respuesta será: `sleep 100` y además impedirá introducir más comandos hasta que no termine ese proceso.

Pasar procesos de primer plano a segundo plano

Primero es necesario parar el proceso si está en ejecución con `Ctrl + Z` Después ya podemos pasarlo a segundo plano con: `bg jobid`

Cambiar prioridad de procesos

IMPORTANTE La prioridad va desde 20 (más alta) a -20 (más baja). A más nivel de prioridad mas lentamente se ejecuta. Solo root puede asignar valores negativos

Asignar prioridad `nice prioridad jobid` Ej: `nice 20 10`

Cambiar prioridad Parámetros:

- g: nivel de prioridad por los miembros del grupo
- u: nivel de prioridad por el usuario
- p: nivel de prioridad para el proceso `renice -n prioridad -p pid` Ej: `renice -n 5 -n 10`

Finalizar procesos `kill -9 PDI` Ej: `kill -9 10` Parámetros:

- 1: reinicia el proceso
- 9: mata el proceso
- 15: termina el proceso
- 17: detiene un proceso
- 19: continua un proceso detenido

Importante Con el comando `killall nombreprograma` puedes eliminar todos los procesos que use ese programa Ej: `killall firefox`