Sistemas Informáticos  
Unidad 05. Actividades no evaluables 03

short line

Autores: Sergi García, Alfredo Oltra

Actualizado Septiembre 2025

Unidad 05. Actividades no evaluables 03

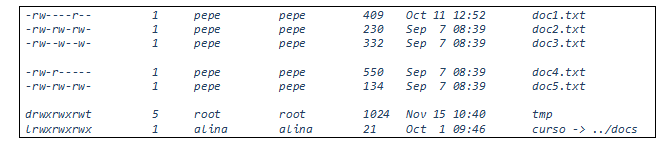
# Ejercicio 01

**Realiza esos ejercicios usando los comandos touch, cat, cd, ls, mkdir, cp, mv, rmdir, rm, grep.**

* Escribe un comando para crear un nuevo fichero llamado names.txt.
* Escribe un comando para ver el contenido de names.txt.
* Escribe un comando para ver el contenido de tu directorio personal en formato largo (permisos, tamaño, fecha, etc.).
* Escribe un comando para ver el contenido del directorio actual en formato largo, mostrando ficheros/directorios ocultos (permisos, tamaño, fecha, etc.).
* Escribe un comando para listar todos los ficheros que terminen en .png y empiecen por ga.
* Escribe un comando para guardar el resultado de un ls en un fichero llamado myLS.txt, **borrando** el contenido anterior.
* Escribe un comando para guardar el resultado de un ls en un fichero llamado myLS.txt, **añadiendo** el resultado al final del fichero existente.
* Escribe un comando para crear un directorio llamado Exercise1 en tu home.
* Escribe un comando para mover todos los ficheros que empiecen por a desde tu home al directorio Exercise1.
* Escribe un comando para cambiar el nombre del directorio Exercise1 a Ex1.
* Escribe un comando para mostrar las líneas de /etc/passwd que contienen la palabra root.
* Borra todos los elementos creados.

# Ejercicio 01

Hemos obtenido este resultado ejecutando ls -l.



* En **modo simbólico**: añade permiso de **ejecución** al propietario de doc1.txt.
* En **modo simbólico**: elimina el permiso de **escritura** para grupo y otros de doc2.txt.
* En **modo octal**: añade permiso de **ejecución** al grupo de doc4.txt.
* En **modo octal**: elimina el permiso de **escritura** para el grupo y los permisos de **lectura y escritura** para otros del fichero doc5.txt.
* Escribe un comando para cambiar **propietario** a Eulogio y **grupo** a Eulogio de todos los ficheros del directorio.

## 

# Ejercicio 03

1. Crea el usuario pepito por línea de comandos.
2. Crea el grupo tic por línea de comandos.
3. Cambia el grupo primario del usuario pepito a tic.

# Ejercicio 04

Resuelve estos ejercicios usando el comando grep.  
 💬 **Interesante**: puedes encadenar comandos grep usando el operador | (pipe).

* Muestra todas las líneas del fichero list.txt que contengan el texto lib.
* Muestra cuántas líneas contienen mp3 en list.txt.
* Muestra qué ficheros dentro del directorio /etc/ contienen la cadena host en su interior.
* Muestra todas las líneas del fichero list.txt que **no** contienen la letra a (mayúscula o minúscula).
* Muestra todas las líneas del fichero list.txt que **no** contienen a (mayúscula o minúscula) **y** que contienen m (minúscula).

💬 **Consejo**: | es un operador que redirige la salida de un comando como entrada de otro. Ejemplo: cat file.txt | sort ordena el contenido de file.txt.

# Ejercicio 05

* Crea una carpeta llamada shared en tu home donde **todo el mundo** tenga **todos** los permisos.
* Crea los grupos office1 y office2.
* Crea los usuarios pedro y pablo. Estos usuarios deben ser miembros del grupo office1.
* Crea los usuarios alba y nerea. Estos usuarios deben ser miembros del grupo office2.
* Como pedro, crea un fichero topsecret.txt que **solo** pedro pueda leer y escribir.
* Como pedro, crea un fichero sales.txt que el dueño y el grupo office1 puedan leer y escribir. Comprueba como pablo si puede realizar esas operaciones.
* Como alba, crea un fichero employ.txt que **todos** los usuarios puedan leer y el grupo office2 pueda leer y escribir. Comprueba que funciona con pedro y nerea.

# Ejercicio 06

**Parte 1:**

**Pregunta 01:** si un usuario tiene permiso de lectura sobre un fichero, pero ese fichero está dentro de un directorio en el que nuestro usuario **no** tiene permiso de ejecución y **sí** tiene permiso de lectura en ese directorio: ¿podría leer el fichero?

**Pregunta 02:** si un usuario tiene permiso de lectura sobre un fichero, pero ese fichero está dentro de un directorio en el que nuestro usuario **no** tiene permiso de lectura y **sí** tiene permiso de ejecución en ese directorio: ¿podría leer el fichero?

**Parte 2:**

Suponiendo que temporalmente (por ejemplo 1 hora) tienes acceso como **root** en una máquina y en esa máquina existe permanentemente un usuario llamado alumno sin permisos en sudoers:

**Pregunta 01:** ¿Cómo podríamos usar el bit SetUID para crear una puerta trasera (backdoor)? (Pista: el fichero /bin/sh puede ser útil).

**Pregunta 02:** ¿Cómo detectaríamos ese tipo de puertas traseras en nuestro sistema? ¿Qué medidas podemos tomar para estar protegidos frente a este tipo de ataque?

# Ejercicio 07

1. Crea en tu home un directorio llamado proyecto\_acl.
2. Dentro de proyecto\_acl crea los usuarios ana y luis (si no existen) y el grupo equipo. Añade ana y luis al grupo equipo.
3. Crea dentro de proyecto\_acl un archivo informe.txt con contenido cualquiera.
4. Usa **ACL** para dar a:  
   * ana permisos **lectura+escritura** sobre informe.txt.
   * al grupo equipo permisos **lectura** sobre informe.txt.
5. Configura **ACL por defecto** en proyecto\_acl de forma que cualquier fichero nuevo creado allí herede permisos: propietario tiene rwx, grupo equipo rw y los demás ---.
6. Crea un nuevo fichero nuevo.txt en proyecto\_acl y comprueba que ha heredado las ACL por defecto.
7. Muestra todas las ACL aplicadas y, finalmente, elimina las ACL de informe.txt y de proyecto\_acl.

Al terminar, publica en el foro qué resultados obtuviste (si funcionó o no) y captura de salida de getfacl.