



UD5. ADMINISTRACIÓN DE UN SO LIBRE II

IES Comercio

UD5. ADMINISTRACIÓN DE UN SO LIBRE II

1. [GESTIÓN DE USUARIOS Y GRUPOS – GESTIÓN DE USUARIOS](#)
2. [GESTIÓN DE USUARIOS Y GRUPOS – COMANDOS SOBRE PERMISOS](#)
3. [GESTIÓN DE PROCESOS](#)
4. [SCRIPTING](#)
5. AUTOMATIZACIÓN DE TAREAS

Avanzar en conceptos sobre administración avanzada de un SO libre mediante comandos (CLI):

UD05.- Administración de un Sistema Operativo Libre II
1.- Gestión de usuarios y grupos
2.- Gestión de procesos
3.- Script de Linux
4.- Automatizar tareas

1. Gestión de usuarios y grupos

1. Control de tareas y procesos
2. Comandos de información del sistema. Monitor.
3. Sucesos y logs del sistema
4. Scripting
5. Crontab

CONTENIDOS Y OBJETIVOS ESPECÍFICOS

UD05.- Administración de un Sistema Operativo Libre II

1.- Gestión de usuarios y grupos

1.1.- Comandos sobre permisos

1.2.- Comandos de gestión de usuarios

2.- Gestión de procesos

3.- Script de Linux

4.- Automatizar tareas

- Crear, modificar y eliminar usuarios del sistema.
- Crear, modificar y eliminar grupos del sistema.
- Conocer los distintos tipos de grupos del sistema.
- Introducir/eliminar usuarios de los grupos.
- Conocer los permisos básicos.
- Conocer los permisos avanzados.
- Conocer la delegación de permisos.
- Conocer los archivos que se ejecutan al loguearse un usuario.

CONTENIDOS Y OBJETIVOS ESPECÍFICOS

UD05.- Administración de un Sistema Operativo Libre II
1.- Gestión de usuarios y grupos
2.- Gestión de procesos
2.1.- Qué son los procesos
2.2.- Comandos de gestión de procesos
2.3.- Comandos para mostrar información de los procesos
2.4.- Comandos para cambiar el estado de los procesos
2.5.- Comandos para cambiar la prioridad de los procesos
2.6.- Comandos para cambiar el plano de los procesos
3.- Script de Linux

- Controlar las tareas o procesos del sistema mediante los comandos oportunos.
- Utilizar los comandos de información del sistema.
- Conocer los sucesos del equipo.

CONTENIDOS Y OBJETIVOS ESPECÍFICOS

UD05.- Administración de un Sistema Operativo Libre II
1.- Gestión de usuarios y grupos
2.- Gestión de procesos
3.- Script de Linux
3.0.- Ejecución de varios comandos
3.1.- ¿Qué es un script?
3.2.- Variables
3.3.- Parámetros
3.4.- Comandos de programación
3.5.- Operadores de comparación
3.6.- Operaciones aritméticas
3.7.- Depuración.
3.8.- Estructuras
3.9.- Funciones
4.- Automatizar tareas

- Codificar secuencias de comandos para tareas complejas

CONTENIDOS Y OBJETIVOS ESPECÍFICOS

UD05.- Administración de un Sistema Operativo Libre II

1.- Gestión de usuarios y grupos

2.- Gestión de procesos

3.- Script de Linux

4.- Automatizar tareas

4.1. Crontab y cron

4.2. Comando at

- Automatizar tareas o procesos mediante los comandos oportunos, ya sean periódicas o puntuales, mediante scripts

GESTIÓN DE USUARIOS Y GRUPOS



COMANDOS DE GESTIÓN DE USUARIOS

Tipos de usuarios

Solo hay tres tipos de usuarios en sistemas GNU/Linux:

- usuario **normal**, el prompt nos mostrará un \$
- usuario **especial**, nologin, usado en los servicios
- **Superusuario, root**, nos mostrará una #

Recordatorio comandos de interés

uname: muestra información del sistema

logname: muestra el usuario que se ha conectado

id: muestra el nombre, número de usuario y grupo/s al que pertenece

whoami: muestra el nombre de usuario



who: muestra los nombres de los usuarios conectados al sistema, sus terminales y la fecha/hora de acceso. En el caso de utilizar la versión desktop de Ubuntu, mostrará los usuarios que han iniciado sesión desde el entorno gráfico



useradd usuario (acrónimo de user add): **añade** usuarios

Nos permite crear un usuario nuevo en el sistema

Si le pasamos el parámetro **--group** nos permite crear el usuario y añadir a un grupo de usuarios

userdel usuario (acrónimo de user delete): **elimina** usuarios

Si le pasamos el parámetro **-r**, borra también toda la información de su directorio personal

adduser nombre_usuario: **añade** un usuario nuevo al sistema, creando un **directorio de trabajo** con el mismo nombre de usuario dentro de la ruta /home y le asigna una **contraseña**

Más cómodo de utilizar que el comando useradd

passwd usuario: cambia la **contraseña** del usuario indicado

groupadd nuevogrupo: para añadir un nuevo grupo de usuarios



groupdel nombre_del_grupo: para eliminar un grupo

delgroup nombre_usuario nombre_grupo: permite eliminar a un usuario de un grupo de usuarios

usermod: cambia varios atributos de usuarios. Todo usuario puede pertenecer a un solo grupo principal y a varios secundarios

Si le pasamos los siguientes parámetros:

-g nombre_grupo usuario: cambia el grupo principal de un usuario

-G nombre_grupo usuario: nos permite añadir un usuario un grupo secundario

Actividad

P14 – Gestión de usuarios I

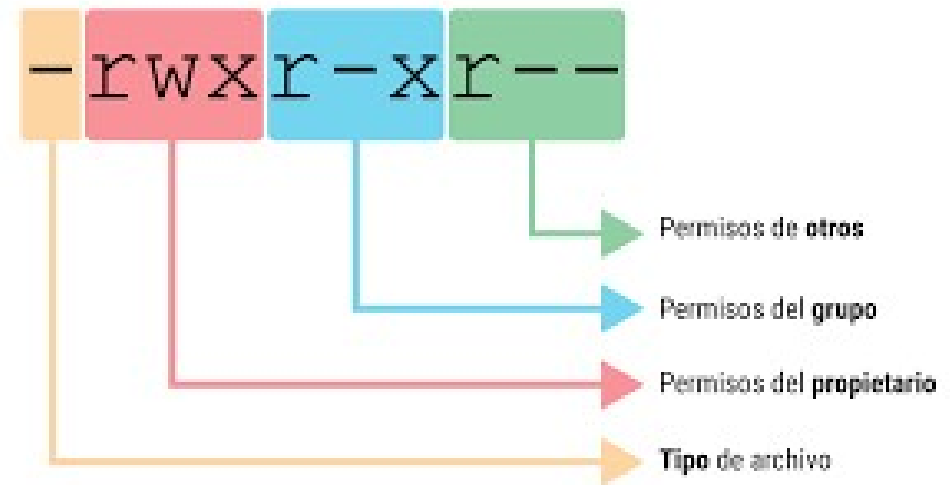
GESTIÓN DE USUARIOS Y GRUPOS



COMANDOS SOBRE PERMISOS

PERMISOS

- Permiso de lectura
- Permiso de escritura
- Permiso de ejecución



r = read (lectura).....(**4**)
w = write (escritura).....(**2**)
x = execute (ejecución)..(**1**)
- = sin permiso.....(**0**)

PERMISOS ASOCIADOS - Directorios

- **Ninguno:** ningún acceso. -
- **Sólo listar archivos:** permiso de ejecución X
- **Acceder a archivos:** permiso de lectura y ejecución R-X
- **Crear y eliminar archivos:** permiso de lectura, escritura y ejecución RWX

Permisos - octal

Número	Binario	Lectura (r)	Escritura (w)	Ejecución (x)
0	000	✗	✗	✗
1	001	✗	✗	✓
2	010	✗	✓	✗
3	011	✗	✓	✓
4	100	✓	✗	✗
5	101	✓	✗	✓
6	110	✓	✓	✗
7	111	✓	✓	✓

CAMBIAR PERMISOS

chmod



`chmod [opción] modo archivo`

- v : muestra un diagnóstico para cada archivo procesado
- c : como verbose pero informando sólo cuando se hace un cambio
- reference=FILE : utiliza el modo de FILE en lugar de los valores de MODE
- R : cambia los permisos recursivamente

CAMBIAR PERMISOS

chmod

chmod [opción] modo archivo

- X = 1
- W = 2
- R = 4

4 + 2 + 1

EJEMPLOS – Permisos modo numérico

1) `chmod 700 nombre_archivo`

2) `chmod -R 755 directorio`

3) `chmod 777 nombre_archivo`

Recomendación general

Asignar permisos 777 es peligroso



Los **permisos más estándar** son:

- **644 para ficheros**
- **755 para directorios**

Esta opción genérica no tiene por qué ser la necesaria para todos los proyectos, pero siempre es una solución bastante apropiada.

CAMBIAR PERMISOS

chmod

chmod [opción] modo archivo



{u,g,o} {+,-,=} {r,w,x}

CAMBIAR PERMISOS

chmod

Modificadores y argumentos para la orden chmod

Quién

u: es el usuario creador.
g: es el grupo al que pertenece el usuario.
o: es el resto de usuarios del equipo.
a: todos los usuarios del sistema

Acción + / -

r: permiso de lectura ($4_{(8)}$)
w: permiso de escritura ($2_{(8)}$)
x: permiso de ejecución ($1_{(8)}$)

EJEMPLOS – Permisos modo simbólico

1) `chmod u=rwx nombre_archivo`

2) `chmod +x /direccion/del/archivo`

3) `chmod -v u+rw /dirección/del/archivo`

4) `chmod -cR g+w /dirección/directorio`

Con **chmod** y **sudo** ahora tienes el poder de cambiar los permisos en casi cualquier archivo.

Esto NO significa que debas hacerlo.



Los permisos fuera del directorio personal están establecidos por una razón!!

Cambiarlos raramente es la solución apropiada a cualquier problema!!

PERMISOS ESPECIALES

Sticky Bit

*Se aplica solo a **directorios** y evita que los usuarios **eliminen o cambien el nombre** de un archivo o directorio a menos que sean **propietarios** de ese archivo o directorio*

PERMISOS ESPECIALES

Sticky Bit

- Modo numérico: valor **octal 1**
- Modo simbólico: dentro de los permisos de "otros":
 - t**: con permiso de ejecución para otros
 - T**: sin permiso de ejecución para otros

EJEMPLOS – Sticky Bit

1) `drwxrwxrwt` 13 root root 4096 ene 24 14:28 /tmp

2) `chmod 1755 Dir1`

`drwxr-xr-t` 2 profe profe 4,0K ene 24 18:46 Dir1

PERMISOS ESPECIALES

Set GID



- ✓ En archivos ejecutables: ejecuta el archivo con los privilegios del grupo
- ✓ En directorios: hace que cada archivo o directorio creado debajo herede el grupo del directorio principal
- Modo numérico: valor **octal 2**
- Modo simbólico: **s** dentro de los permisos de “grupo”

EJEMPLOS – Set GID

```
1) ls -ld /tmp
```

```
-rwxr-Sr-X 1 profe profe 33 ene 24 10:36 test.sh
```

```
2) chmod g+S test.sh
```

```
ls -l test.sh
```

```
-rwxr-Sr-X 1 profe root 33 ene 24 10:36 test.sh
```

PERMISOS ESPECIALES

Set UID



Solo en archivos ejecutables: ejecuta el archivo con los permisos del propietario

- Modo numérico: valor **octal 4**
- Modo simbólico: **s** dentro de los permisos de “usuario”

EJEMPLOS – Set UID

```
ls -l test.sh
```

```
-rwsr-xr-x 1 profe profe 33 ene 24 10:46 test
```

COMBINACIÓN DE PERMISOS ESPECIALES

```
chmod 6755 test.sh
```

```
ls -lh test.sh
```

```
-rwsr-sr--X 1 profe profe 66 ene 17:29 tst.sh
```

④+②+①

s s t

CAMBIAR PERMISOS

chown: cambia el propietario de un archivo

`chown [opciones] usuario[:grupo] archivo(s) o directorio(s)`

chgrp: cambia el grupo

`chgrp [opciones] grupo archivo(s) o directorio(s)`

Actividad

P15 – Gestión de usuarios II