



## Módulo 2: Manipulación de archivos

### Objetivos

- Crear archivos vacíos o actualizar fechas con `touch`.
- Mover y renombrar archivos y directorios con `mv`.
- Copiar archivos y directorios con `cp`.
- Eliminar archivos y directorios de forma segura con `rm`.
- Ver el contenido completo de un archivo con `cat`.
- Navegar por archivos largos con `less`.
- Mostrar el principio o el final de un archivo con `head` y `tail`.

### Contenido teórico

#### Crear archivos vacíos: `touch`

El comando `touch` sirve para:

- Crear un archivo nuevo y vacío (si no existe).
- Actualizar la marca de tiempo de un archivo existente (como si se hubiera modificado).

Bash 1: Ejemplos de `touch`

```
touch archivo.txt          # Crea "archivo.txt" vacío
```



## Mover y renombrar: **mv**

**mv** (move) mueve archivos o directorios de un lugar a otro. Si el destino es una carpeta, el archivo se traslada; si el destino es un nombre (dentro de la misma carpeta), se renombra.

### Bash 2: Ejemplos de **mv**

```
mv arch.txt carpeta/          # Mueve arch.txt dentro de carpeta/  
mv arch.txt nvo_nombre.txt    # Renombra dentro de la misma carpeta  
mv carpeta/ otra_carpeta/     # Mueve la carpeta entera  
mv archivo.txt ../            # Mueve al directorio padre
```

## Copiar archivos y directorios: **cp**

**cp** copia archivos. Para copiar directorios (y su contenido) se necesita la opción **-r** (recursiva). Opciones útiles:

- **-i**: pregunta antes de sobrescribir (interactivo).
- **-v**: muestra qué se está copiando (verbose).
- **-u**: copia solo si el origen es más reciente que el destino (update).
- **-r**: copia recursiva (para directorios).

### Bash 3: Ejemplos de **cp**

```
cp original.txt copia.txt      # Copia archivo  
cp -r docs/ backup_docs/      # Copia directorio completo  
cp -iv archivo.txt carpeta/    # Copia archivo a carpeta  
preguntando antes de sobrescribir
```

## Eliminar archivos y directorios: **rm** y **rmdir**

**rm** borra archivos. ¡Cuidado! No va a la papelera, se pierden para siempre.

- **rm archivo** – borra un archivo.
- **rm -r carpeta** – borra recursivamente una carpeta y todo su contenido.
- **rm -i archivo** – pregunta antes de borrar (modo interactivo, muy recomendable).



- `rmdir carpeta` – borra **solo** carpetas vacías (más seguro, pero menos usado).

#### Bash 4: Ejemplos de `rm`

```
rm basura.txt           # Borra archivo
rm -i documento.txt    # Pregunta antes de borrar
rm -r proyecto_antiguo/ # Borra la carpeta y todo lo que contiene
rmdir carpeta_vacia/    # Solo funciona si está vacía
```

#### Advertencia sobre `rm`

`rm` es irreversible. Nunca ejecutes `rm -rf /` ni `sudo rm -rf /*`, ya que borraría todo el sistema. Siempre verifica dos veces el comando antes de pulsar Enter. Usa `-i` cuando tengas dudas.

## Ver contenido completo: `cat`

`cat` (concatenate) muestra el contenido de uno o varios archivos en la terminal. Es útil para archivos pequeños.

#### Bash 5: Ejemplos de `cat`

```
cat archivo.txt          # Muestra el contenido
cat archivo1.txt archivo2.txt # Muestra uno detrás del otro
```

## Navegar por archivos largos: `less`

Para archivos grandes, `less` abre una interfaz que permite desplazarse y buscar. Atajos:

- Flechas  $\uparrow/\downarrow$  o `PgUp/PgDn` – moverse.
- `/palabra` – buscar hacia adelante.
- `?palabra` – buscar hacia atrás.
- `n` – siguiente coincidencia.
- `N` – coincidencia anterior.
- `g` – ir al principio del archivo.
- `G` – ir al final.



- **q** – salir.

### Bash 6: Uso de **less**

```
less /var/log/syslog # Abre el log del sistema
```

## Principio y final: **head** y **tail**

- **head** muestra las primeras líneas (por defecto 10).
- **tail** muestra las últimas líneas (por defecto 10).
- Con **-n** se especifica el número de líneas.
- **tail -f** (follow) muestra en tiempo real las líneas que se agregan a un archivo (ideal para logs).

### Bash 7: Ejemplos de **head** y **tail**

```
head -n 5 archivo.txt # Primeras 5 líneas  
tail -n 20 archivo.txt # Últimas 20 líneas  
tail -f /var/log/syslog # Muestra nuevas líneas en vivo (Ctrl+C  
para salir)
```

#### Tip 4: Sintaxis abreviada de **head** y **tail**

En lugar de escribir **head -n 5 archivo.txt** o **tail -n 5 archivo.txt**, puedes usar directamente:

```
head -5 archivo.txt
```

```
tail -5 archivo.txt
```

El número de líneas se coloca inmediatamente después del guion, omitiendo la **-n**.

La misma sintaxis funciona con **head** y **tail**. ¡Es más rápida y muy común en la práctica!

## Recordatorio de tips anteriores

- **TAB**: autocompleta comandos y rutas.



- **Flechas** ↑/↓: historial de comandos.
- **Expansión de llaves:** `mkdir -p proyecto/{src,doc,backup}` (visto en el Módulo 1).

## Ejercicios prácticos

Resuelve en tu terminal:

1. Crea un archivo vacío llamado `notas.txt` con `touch`.
2. Usa `mv` para renombrar `notas.txt` a `apuntes.txt`.
3. Crea una carpeta `mis_docs` y mueve `apuntes.txt` dentro de ella.
4. Copia `apuntes.txt` a `apuntes_backup.txt` dentro de la misma carpeta.
5. Con `cat`, muestra el contenido de `apuntes.txt` (si está vacío, primero agrega algo usando un editor o el comando `echo`, que veremos después).
6. Muestra las primeras 3 líneas de `apuntes.txt` con `head` (si el archivo tiene menos de 3 líneas, `head` mostrará todas).
7. Muestra las últimas 2 líneas con `tail`.
8. Abre `apuntes.txt` con `less`, practica moverte con flechas, busca una palabra con `/`, y sal con `q`.
9. Crea varios archivos con nombre `error1.log`, `error2.log`, `error3.log` usando `touch` (uno por uno o con expansión de llaves: `touch error{1,2,3}.log`).
10. Borra uno de esos archivos con `rm` (puedes usar `-i` para practicar confirmación).
11. (Desafío) Crea una carpeta `prueba_rm` con varios archivos dentro. Luego intenta borrar la carpeta con `rmdir` (deberá fallar porque no está vacía). Finalmente bórrala con `rm -r`.
12. (Opcional - Avanzado) Observa cómo se actualiza un archivo en tiempo real usando `tail -f` sobre tu propio historial de comandos (`.bash_history`):



- Abre **dos terminales** (ventanas de terminal independientes).
- En la primera terminal, ejecuta:

```
tail -f ~/.bash_history
```

- En la segunda terminal, escribe cualquier comando, por ejemplo:

```
echo "Hola, esto es una prueba"  
exit
```

- Vuelve a la primera terminal. Deberías ver que el comando **echo** y **exit** recién escrito aparece automáticamente en pantalla.
- Para salir de **tail -f**, presiona **Ctrl + C**.

## Referencias

- The Linux Command Line (TLCL) – Capítulos 5 y 6.
- The Bash Guide – Sección sobre manipulación de archivos.

## Resumen

En este módulo aprendiste a:

- Crear archivos vacíos con **touch**.
- Mover y renombrar con **mv**.
- Copiar archivos (y directorios con **-r**) usando **cp**.
- Eliminar de forma segura con **rm** (y **-i**, **-r**), y la diferencia con **rmdir** (solo para carpetas vacías).
- Ver contenidos con **cat** (archivos pequeños), navegar con **less**, y extraer principio/fin con **head** y **tail**, incluyendo el modo **-f** para seguir logs.

Ahora puedes manipular archivos con confianza. En el próximo módulo veremos redirecciones, tuberías y comandos de filtrado.