



Taller Práctico: Dominando el Shell Bash



Ejercicio 1: Protección contra sobrescritura (`noclobber`)

- **Objetivo:** Evitar borrar el contenido de un archivo importante por un error al usar el redireccionador `>`.
- **Paso a paso:**
 1. Crea un archivo con contenido: `echo "Datos importantes" > notas.txt`
 2. Activa la protección: `set -o noclobber`
 3. Intenta sobrescribirlo: `echo "Nuevos datos" > notas.txt`
- **Explicación:** Al activar `noclobber`, Bash impedirá que el operador `>` destruya un archivo existente. Recibirás un mensaje de error tipo: *cannot overwrite existing file.* 🛡️



Ejercicio 2: Forzar sobrescritura con el operador `>|`

- **Objetivo:** Aprender a ignorar la protección `noclobber` de forma puntual.
- **Paso a paso:**
 1. Con `noclobber` activo, ejecuta: `echo "Actualización forzada" >| notas.txt`
 2. Comprueba el contenido: `cat notas.txt`
- **Explicación:** El operador `>|` es la "llave maestra". Le dice a Bash: "Sé que el archivo existe y que tengo una protección puesta, pero quiero sobrescribirlo de todos modos". 🔑



Ejercicio 3: Desactivar opciones del shell

- **Objetivo:** Volver al comportamiento por defecto de la terminal.
- **Paso a paso:**
 1. Ejecuta: `set +o noclobber`
 2. Intenta sobrescribir: `echo "Vuelvo a la normalidad" > notas.txt`
- **Explicación:** En el comando `set`, el signo `-` activa una opción y el signo `+` la desactiva. Es un poco contraintuitivo, pero esencial recordarlo. ↺



Ejercicio 4: Gestión del historial con `history`

- **Objetivo:** Visualizar los comandos ejecutados recientemente.
- **Paso a paso:**
 1. Ejecuta: `history`
 2. Para ver solo los últimos 5: `history 5`

- **Explicación:** El comando `history` muestra una lista numerada de los comandos que has introducido. Es la base para recuperar trabajo previo. 📜

Ejercicio 5: Ejecución rápida desde el historial (!)

- **Objetivo:** Ejecutar un comando anterior sin tener que escribirlo de nuevo.
- **Paso a paso:**
 1. Busca un número en tu historial (ej: el 42).
 2. Ejecuta: `!42`
 3. Para repetir el último comando: `!!`
- **Explicación:** El símbolo `!` seguido de un número lanza el comando asociado a esa posición. `!!` es muy útil cuando olvidas poner `sudo` (puedes hacer `sudo !!`). ⚡

Ejercicio 6: Búsqueda interactiva en el historial (Ctrl + R)


- **Objetivo:** Encontrar un comando complejo que escribiste hace tiempo.
- **Paso a paso:**
 1. Pulsa las teclas `Ctrl` y `R` al mismo tiempo.
 2. Empieza a escribir una palabra (ej: `tar` o `ssh`).
 3. Pulsa `Ctrl` + `R` varias veces para navegar por los resultados.
 4. Pulsa `Enter` para ejecutar o las flechas para editar.
- **Explicación:** Es la forma más profesional de buscar. Realiza una búsqueda inversa hacia atrás en el historial mientras escribes. 🔍

Ejercicio 7: Recuperar el último argumento (Alt + .)

- **Objetivo:** Reutilizar la ruta o nombre de archivo del comando anterior.
- **Paso a paso:**
 1. Crea un directorio largo: `mkdir /home/curso/practicas/linux/servidores`
 2. Escribe `cd` (con espacio al final) y pulsa `Alt` + `.` (punto).
- **Explicación:** Este atajo (o `ESC` y luego `.`) pega automáticamente el último parámetro del comando anterior. Te ahorra escribir rutas larguísimas. 🖱️

Ejercicio 8: Creación de Alias temporales

- **Objetivo:** Crear un comando corto personalizado para una sesión actual.
- **Paso a paso:**
 1. Ejecuta: `alias lls='ls -lAhF --color=auto'`
 2. Prueba tu nuevo comando: `lls`

- **Explicación:** Los alias permiten renombrar comandos largos. Este alias en concreto muestra archivos ocultos, detalles y colores con solo tres letras. 

Ejercicio 9: Persistencia de Alias en el fichero `.bashrc`


- **Objetivo:** Hacer que tus alias y configuraciones de `set` no se borren al cerrar la terminal.

- **Paso a paso:**

1. Edita el archivo de configuración: `vi ~/.bashrc`
2. Ve al final del documento e inserta:

```
alias c='clear'
alias update='sudo dnf update' # (En Rocky) o sudo apt update (En
Ubuntu)
set -o noclobber
```

3. Guarda y sal (`:wq`).

- **Explicación:** `.bashrc` es un script que se ejecuta automáticamente cada vez que abres una nueva terminal. 

Ejercicio 10: Aplicar cambios sin cerrar sesión (`source`)

- **Objetivo:** Activar los cambios recién hechos en el `.bashrc` inmediatamente.

- **Paso a paso:**

1. Ejecuta: `source ~/.bashrc`
2. Prueba tus nuevos alias: `c` (debería limpiar la pantalla).

- **Explicación:** Por defecto, los cambios en `.bashrc` solo se aplican en terminales nuevas. El comando `source` obliga al shell actual a leer el archivo y aplicar la configuración al instante. 