

## 2-SCRIPTS

Crea un script que se llame `borra.sh`, que borre la pantalla y te diga la fecha de hoy. – ¿Cómo cambias los permisos para que se ejecute? – ¿Cómo lo ejecutas?

USO comando para el hacer el script: `nano borra.sh`

-dentro de el:

```
#!/bin/bash
clear
echo "La fecha de hoy es: $(date)"
```

-ahora le doy permisos y ejecuto el script  
`chmod +x borra.sh`

para ejecutar hice: `./borra.sh`

```
La fecha de hoy es: Tue Nov 11 09:33:51 AM CET 2025
egibide@Bryan:~$
```

---

### EJERCICIO SCRIPTS 1:

- Cambia el valor de la variable `PATH` para que te incluya tu directorio personal.

```
egibide@Bryan:~$ export PATH=$PATH:$HOME
egibide@Bryan:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib:/mnt/c/Program Files/WindowsApps/MicrosoftCorporationI
I.WindowsSubsystemForLinux_2.6.1.0_x64__8wekyb3d8bbwe:/mnt/c/Users/kimbe/AppData
/Roaming/MobaXterm/slash/mx86_64b/bin:/mnt/c/WINDOWS:/mnt/c/WINDOWS/system32:/
mnt/c/windows/system32:/mnt/c/windows:/mnt/c/windows/System32/Wbem:/mnt/c/window
s/System32/WindowsPowerShell/v1.0:/mnt/c/Program Files (x86)/NVIDIA Corporation
/PhysX/Common:/mnt/c/Program Files/NVIDIA Corporation/NVIDIA app/NvDLISR:/mnt/c/
WINDOWS/system32:/mnt/c/WINDOWS:/mnt/c/WINDOWS/System32/Wbem:/mnt/c/WINDOWS/Syst
em32/WindowsPowerShell/v1.0:/mnt/c/Program Files/Git/cmd:/mnt/c/Users/kimbe/App
Data/Local/Microsoft/WindowsApps:/mnt/c/Users/kimbe/AppData/Local/GitHubDesktop/
bin:/mnt/c/Users/kimbe/AppData/Local/Programs/Microsoft VS Code/bin:/mnt/c/WINDO
WS/sysnative:/home/egibide
egibide@Bryan:~$
```

- A continuación crea un shell script, `param.sh`, que muestre los parámetros que ha recibido, cuántos parámetros son, el nombre del shell script y el PID del proceso.

nano [param.sh](#)

```
#!/bin/bash

echo " Parámetros recibidos: $@"
echo " Número de parámetros: $# "
echo " Nombre del script: $0 "
echo " PID del proceso: $$ "
```

le doy permisos: **chmod +x param.sh**

```
egibide@Bryan:~$ ./param.sh buenas noches mi estimada
Parámetros recibidos: buenas noches mi estimada
Número de parámetros: 4
Nombre del script: ./param.sh
PID del proceso: 31084
egibide@Bryan:~$
```

## EJERCICIO SCRIPTS 2:

- Crea un shell script, fich.sh, que te pregunte el nombre de un fichero. Si el fichero existe, te debe mostrar información sobre él en formato largo. Si no existe o no es un fichero, te mostrará un mensaje de que no existe.

nano [fich.sh](#)

```
GNU nano 8.4 fich.sh
#!/bin/bash

read -p "Introduce el nombre del fichero: " fichero

if [ -f "$fichero" ]; then
    echo "☑ El fichero '$fichero' existe. Información detallada:"
    ls -l "$fichero"
else
    echo "✗ El fichero '$fichero' no existe o no es un fichero regular."
fi
```

ahora le doy permisos: **chmod +x fich.sh**

Lo pruebo:

```
egibide@Bryan:~$ ./fich.sh
Introduce el nombre del fichero: borra.sh
☑ El fichero 'borra.sh' existe. Información detallada:
-rwxrwxr-x 1 egibide egibide 55 Nov 11 09:36 borra.sh
```

### EJERCICIO SCRIPTS 3:

- Crea un shell script, fichodir.sh, que reciba una serie de parámetros por la línea de comandos, los muestre y te diga si son ficheros o directorios.

#### nano [fichodir.sh](#)

```
#!/bin/bash

if [ $# -eq 0 ]; then
    echo "✗ No se ha introducido ningún parámetro."
    echo "Uso: ./fichodir.sh nombre1 nombre2 ..."
    exit 1
fi

echo " Parámetros recibidos: $@"
echo "-----"

# Recorremos todos los parámetros
for elemento in "$@"
do
    if [ -f "$elemento" ]; then
        echo " '$elemento' es un fichero."
    elif [ -d "$elemento" ]; then
        echo " '$elemento' es un directorio."
    else
        echo " '$elemento' no existe o no es ni fichero ni directorio."
    fi
done
```

#### chmod +x fichodir.sh

```
egibide@Bryan:~$ ./fichodir.sh borra.sh /home /etc inventado.txt
Parámetros recibidos: borra.sh /home /etc inventado.txt
-----
'borra.sh' es un fichero.
'/home' es un directorio.
'/etc' es un directorio.
'inventado.txt' no existe o no es ni fichero ni directorio.
```

### EJERCICIO SCRIPTS 4:

- Crea un shell script, reciente.sh, que te pregunte el nombre de un fichero. Si existe, para cada entrada del directorio personal, te dirá si es más reciente o no que el fichero que has escrito.
- NOTA, para comparar si es mas reciente \$var1 -nt \$var2

LE DOY PERMISOS:

#### chmod x+ reciente.sh

```

read -p "Introduce el nombre de un fichero: " fichero

if [ ! -f "$fichero" ]; then
    echo "El fichero '$fichero' no existe."
    exit 1
fi

for entrada in "$HOME"/*; do
    if [ "$entrada" -nt "$fichero" ]; then
        echo "- $entrada es MÁS RECIENTE que $fichero"
    else
        echo "* $entrada NO es más reciente que $fichero"
    fi
done

```

### EJERCICIO SCRIPTS 5:

- Mira en tu directorio personal los archivos ocultos para ver cuales de los scripts de inicio de sesión tienes.

```

egibide@Bryan:~$ ls -a
.  .bash_history  .bashrc  fichodir.sh  .lessht  param.sh  reciente.sh  .Xauthority
.. .bash_logout  borra.sh  fich.sh     .local    .profile  .sudo_as_admin_successful

```

### EJERCICIO SCRIPTS 6:

- Modifica los ficheros .profile y .bashrc de tu directorio personal y los ficheros /etc/profile y /etc/bashrc añadiéndoles una línea al inicio que muestre por pantalla el nombre del fichero que se va a ejecutar.
- Para ver el orden en que se ejecutan los ficheros inicia una nueva sesión en la primera terminal virtual.

```

Ejecutando: /etc/profile
Ejecutando: /etc/bash.bashrc
Ejecutando .profile
Ejecutando .bashrc
egibide@Bryan:~$

```

### • EJERCICIO SCRIPTS 7:

- Modifica los ficheros `.profile` y `.bash_logout` de tu directorio personal de manera que cada vez que se inicie una sesión se escriba en un fichero llamado `.movimientos` (si se entra, si se sale del sistema o si se inicia una nueva sesión) el nombre del fichero shell script que se está ejecutando y la fecha.

```
Ejecutando: /etc/profile
Ejecutando: /etc/bash.bashrc
Ejecutando .profile
Ejecutando .bashrc
egibide@Bryan:~$ sudo cat .movimientos
[sudo] password for egibide:
2025-11-25 08:43:36 - Entrada al sistema - Ejecutando: -bash
2025-11-25 08:44:50 - Entrada al sistema - Ejecutando: -bash
2025-11-25 08:45:48 - Salida del sistema - Ejecutando: -bash
2025-11-25 08:45:53 - Entrada al sistema - Ejecutando: -bash
egibide@Bryan:~$
```