

Files

Search

hacker-lab

test

safe_mcd.sh

agenda.txt

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

linux1_shell.md

linux2_commands.md

linux3_scripting.md

mcd.sh

mkcd.sh

comando ejecutado o la última palabra en un comando anterior. comando fue cat file.txt)

A continuación, un script en bash que utiliza varios de los símbolos listados

Ejemplo #4: vasr.sh

¡Vamos a explorar un script que aprovecha al máximo los símbolos especiales en bash! Este script ilustra cómo utilizar diversos símbolos para obtener información valiosa sobre el entorno y los comandos ejecutados. Desde el nombre del script hasta los detalles del último comando en segundo plano, cada línea ofrece una visión de cómo bash maneja los argumentos, los códigos de salida y mucho más. Observa cómo se aplican estos conceptos en la práctica:

```
#!/bin/bash

echo "Nombre del script: $0"
echo "Primer argumento: $1"
echo "Segundo argumento: $2"
echo "Número de argumentos: $#"
```

~/workspace/test/test/test:safe_mcd_test

~/workspace\$ chmod +x mkcd.sh

~/workspace\$./mkcd.sh test

~/workspace\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ mcd test

mcd: command not installed, but was located via Nix.

package: mtools 4.0.41 Utilities to access MS-DOS disks

Would you like to run mcd from Nix and add it to your project? [Yn]: y

mtools 4.0.41 Utilities to access MS-DOS disks

Adding mtools to replit.nix

success

Can't open /dev/fd0: No such file or directory

Cannot initialize 'A:'

~/workspace/test\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ source ./mcd.sh

~/workspace/test\$ mcd test

~/.../test/test\$ code safe_mcd.sh

bash: sorce: command not found

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ safe_mcd test

~/.../test/test\$

Files

Search

hacker-lab

test

safe_mcd.sh

agenda.txt

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

1 safe_mcd() {

2 if [-z "\$1"]; then

3 echo "Error: Debes proporcionar un nombre de directorio."

4 return 1

5 fi

6 mkdir -p "\$1" && cd "\$1" || echo "No se pudo cambiar al

7 directorio."

8 }

~/workspace/test/test/test:safe_mcd_test

~/workspace\$ chmod +x mkcd.sh

~/workspace\$./mkcd.sh test

~/workspace\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ mcd test

mcd: command not installed, but was located via Nix.

package: mtools 4.0.41 Utilities to access MS-DOS disks

Would you like to run mcd from Nix and add it to your project? [Yn]: y

mtools 4.0.41 Utilities to access MS-DOS disks

Adding mtools to replit.nix

success

Can't open /dev/fd0: No such file or directory

Cannot initialize 'A:'

~/workspace/test\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ source ./mcd.sh

~/workspace/test\$ mcd test

~/.../test/test\$ code safe_mcd.sh

~/.../test/test\$ source safe_mcd.sh

bash: sorce: command not found

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ safe_mcd test

~/.../test/test\$

Files

Search

hacker-lab

test

safe_mcd.sh

agenda.txt

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

hello.txt

1 #!/bin/bash

2 mkdir -p "\$1"

3 cd "\$1"

~/workspace/test/test/test:safe_mcd_test

~/workspace\$ chmod +x mkcd.sh

~/workspace\$./mkcd.sh test

~/workspace\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ mcd test

mcd: command not installed, but was located via Nix.

package: mtools 4.0.41 Utilities to access MS-DOS disks

Would you like to run mcd from Nix and add it to your project? [Yn]: y

mtools 4.0.41 Utilities to access MS-DOS disks

Adding mtools to replit.nix

success

Can't open /dev/fd0: No such file or directory

Cannot initialize 'A:'

~/workspace/test\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ source ./mcd.sh

~/workspace/test\$ mcd test

~/.../test/test\$ code safe_mcd.sh

~/.../test/test\$ source safe_mcd.sh

bash: sorce: command not found

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ safe_mcd test

~/.../test/test\$

Files

Search

hacker-lab

test

safe_mcd.sh

agenda.txt

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

hello.txt

1 mcd() {

2 mkdir -p "\$1"

3 cd "\$1"

4 }

~/workspace/test/test/test:safe_mcd_test

~/workspace\$ chmod +x mkcd.sh

~/workspace\$./mkcd.sh test

~/workspace\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ mcd test

mcd: command not installed, but was located via Nix.

package: mtools 4.0.41 Utilities to access MS-DOS disks

Would you like to run mcd from Nix and add it to your project? [Yn]: y

mtools 4.0.41 Utilities to access MS-DOS disks

Adding mtools to replit.nix

success

Can't open /dev/fd0: No such file or directory

Cannot initialize 'A:'

~/workspace/test\$ source mcd.sh

bash: mcd.sh: No such file or directory

~/workspace/test\$ source ./mcd.sh

~/workspace/test\$ mcd test

~/.../test/test\$ code safe_mcd.sh

~/.../test/test\$ source safe_mcd.sh

bash: sorce: command not found

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ source safe_mcd.sh

~/.../test/test\$ safe_mcd test

~/.../test/test\$

Files

Search

foca

hacker-lab

scanner.sh

test

test

safe_mcd.sh

agenda.txt

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nom...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

hello.txt

linux1_shell.md

linux2_commands.md

linux3_scripting.md

mcd.sh

mcd.sh


linux3_scripting.md

Linea 6:

echo \$SHELLOPTS

set -o

Explanation

Taller

Taller: Expansión de Llaves en Bash

Objetivos:

- **Verificar la comprensión:** Comprueba que los estudiantes entiendan los conceptos básicos de la expansión de llaves.
- **Aplicación práctica:** Evalúa la capacidad de aplicar la expansión de llaves en escenarios del mundo real.
- **Identificar mejoras:** Destaca posibles áreas de mejora en la comprensión del tema.

Ejercicios:

Shell

~/workspace/test/test: set -o

~/.../test/test\$ echo {1..10..2}

1 3 5 7 9

~/.../test/test\$ echo {a..z..5}

a f k p u z

~/.../test/test\$ mkdir carpeta{1..10}_ver{A..B}

~/.../test/test\$ set +B

~/.../test/test\$ echo {1..10}

{1..10}

~/.../test/test\$ set -B

~/.../test/test\$ echo \$SHELLOPTS

braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor

~/.../test/test\$ set -o

allexport off

braceexpand on

emacs on

errexit off

errtrace off

functrace off

hashall on

histexpand on

history on

ignoreeof off

interactive-comments on

keyword off

monitor on

noclobber off

noexec off

noglob off

nolog off

notify off

nounset off

onecmd off

physical off

pipefail off

posix off

privileged off

verbose off

vi off

xtrace off

~/.../test/test\$

Files

Search

foca

hacker-lab

scanner.sh

test

test

safe_mcd.sh

agenda.txt

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nom...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

hello.txt

linux1_shell.md

linux2_commands.md

linux3_scripting.md

mcd.sh

mcd.sh

linux3_scripting.md

Taller

Taller: Expansión de Llaves en Bash

Objetivos:

- **Verificar la comprensión:** Comprueba que los estudiantes entiendan los conceptos básicos de la expansión de llaves.
- **Aplicación práctica:** Evalúa la capacidad de aplicar la expansión de llaves en escenarios del mundo real.
- **Identificar mejoras:** Destaca posibles áreas de mejora en la comprensión del tema.

Ejercicios:

Shell

~/workspace/test/test/test: shuf -e {a..z}{a..z}{a..z} -n 4

hashall on

histexpand on

history on

ignoreeof off

interactive-comments on

keyword off

monitor on

noclobber off

noexec off

noglob off

nolog off

notify off

nounset off

onecmd off

physical off

pipefail off

posix off

privileged off

verbose off

vi off

xtrace off

~/.../test/test\$ touch archivo{1..10}.txt

~/.../test/test\$ mkdir proyecto_{A..E}

~/.../test/test\$ pd

/home/runner/workspace/test/test/test

~/.../test/test\$ touch {1..3}{a..c}.{txt,pdf}

~/.../test/test\$ mkdir -p projects/{dev,prod}/{config{1..3},logs}

~/.../test/test\$ echo {2..20..2}

2 4 6 8 10 12 14 16 18 20

~/.../test/test\$ echo {b..z..2}

b d f h j l n p r t v x z

~/.../test/test\$ echo 2024-01-{01..10}

2024-01-01 2024-01-02 2024-01-03 2024-01-04 2024-01-05 2024-01-06 2024-01-07 2024-01-08 2024-01-09 2024-01-10

~/.../test/test\$ shuf -e {a..z}{a..z}{a..z} -n 4

cet

vll

gux

qoy

~/.../test/test\$

Files

Search

archivo_2.txt
archivo_3.txt
archivo_4.txt
archivo_5.txt
archivo_6.txt
archivo_7.txt
archivo_8.txt
archivo_9.txt
archivo_10.txt
docninja.sh
genesis.sh
vars.sh
safe_mcd.sh
agenda.txt
banco.txt
colores.txt
contactos.txt
estudiantes_por_id...
estudiantes_por_no...
estudiantes_por_no...
estudiantes.txt
frutas.txt
hello.txt
linux1_shell.md
linux2_commands...
linux3_scripting.md

linux3_scripting.md

Actividad 4: Ejecución Condicional de Scripts

Solución:

El siguiente script cambia su comportamiento basado en los modos de operación de Bash y los parámetros pasados:

```
#!/bin/bash

# Verificar si Bash está en modo de depuración
if [[ $- == *x* ]]; then
    echo "Modo de depuración activado"
fi

# Verificar el número de argumentos
if [[ $# -eq 0 ]]; then
    echo "No se pasaron argumentos."
elif [[ $# -eq 1 && $1 == "start" ]]; then
    echo "Iniciando proceso en segundo plano..."
    sleep 20 &
    echo "Proceso iniciado con PID: $!"
else
    echo "Argumentos no válidos o no se reconocen."
fi
```

Explicación:

- [[\$- == *x*]] comprueba si el modo de depuración (x) está activado.
- [[\$# -eq 0]] verifica si no se pasaron argumentos al script.
- [[\$1 == "start"]] verifica si el primer argumento es start y si es así, inicia un

Shell

~/workspace: if [[\$- == *x*]]; then echo "\Modo de depuración activado"; fi
echo "# Creado el: \$(date)" >> "\$file"
echo "# Descripción: Script principal para el módulo correspondiente." >> "\$file"
file
echo "" >> "\$file"
done
bash: !/bin/bash: event not found
~/workspace\$ #!/bin/bash

Verificar los modos en que Bash está operando
echo "Modos actuales de Bash: \$-"

Ejecutar un comando en segundo plano
sleep 10 &
echo "PID del último comando en segundo plano: \$!"

Imprimir todos los argumentos pasados al script
echo "Argumentos pasados: \$@"
Modos actuales de Bash: htmBts
PID del último comando en segundo plano: 317
Argumentos pasados:
~/workspace\$ #!/bin/bash

Verificar si Bash está en modo de depuración
if [[\$- == *x*]]; then
 echo "Modo de depuración activado"
fi

Verificar el número de argumentos
if [[\$# -eq 0]]; then
 echo "No se pasaron argumentos."
elif [[\$# -eq 1 && \$1 == "start"]]; then
 echo "Iniciando proceso en segundo plano..."
 sleep 20 &
 echo "Proceso iniciado con PID: \$!"
else
 echo "Argumentos no válidos o no se reconocen."
fi
No se pasaron argumentos.
~/workspace\$


Files

Search

foca
hacker-lab
scanners.sh
test
test
safe_mcd.sh
agenda.txt
banco.txt
colores.txt
contactos.txt
estudiantes_por_id_de...
estudiantes_por_nom...
estudiantes_por_not...
estudiantes.txt
frutas.txt
hello.txt
linux1_shell.md
linux2_commands.md
linux3_scripting.md
mcd.sh
mkcd.sh

linux3_scripting.md

\$0, y \$1, y la creación de estructuras complejas de directorios y archivos.

Taller

Taller Avanzado de Scripting en Bash

Objetivos:

- Aplicar la expansión de llaves para la creación eficiente de múltiples archivos y directorios.
- Utilizar y comprender el comportamiento de símbolos especiales como \$-, \$0, \$1 en scripts complejos.
- Desarrollar scripts que automaticen tareas complejas, combinando lo aprendido en módulos anteriores.

Material Necesario: Ordenadores con Bash instalado, acceso a internet, editor de texto.

Parte 1: Expansión de Llaves y Estructuras Complejas

Actividad 1: Creación de Estructuras de Directorios

1. Objetivo: Crear una estructura de directorios organizada para un proyecto ficticio de software.
2. Instrucciones:

- Crear una estructura de directorios para un proyecto de desarrollo con las siguientes características:

Shell

~/workspace/test/test/test: chmod +x docninja.sh
interactive-comments on
keyword off
monitor on
noclobber off
noexec off
noglob off
nolog off
notify off
nounsset off
onecmd off
physical off
pipefail off
posix off
privileged off
verbose off
vi off
xtrace off
~/.../test/test\$ touch archivo {1..10}.txt
~/.../test/test\$ mkdir proyecto {A..E}
~/.../test/test\$ pwd
/home/runner/workspace/test/test/test
~/.../test/test\$ touch {1..3}{a..c}.txt.pdf
~/.../test/test\$ mkdir -p projects/{dev,prod}/{config{1..3},logs}
~/.../test/test\$ echo {2..20..2}
2 4 6 8 10 12 14 16 18 20
~/.../test/test\$ echo {b..z..2}
b d f h j l n p r t v x z
~/.../test/test\$ echo 2024-01-{01..10}
2024-01-01 2024-01-02 2024-01-03 2024-01-04 2024-01-05 2024-01-06 2024-01-07 2024-01-08 2024-01-09 2024-01-10
~/.../test/test\$ shuf -e {a..z}{a..z}{a..z} -n 4
oel
vtl
gux
qoy
~/.../test/test\$ code genesis.sh
~/.../test/test\$ chmod +x genesis.sh
~/.../test/test\$ code docninja.sh
~/.../test/test\$ chmod +x docninja.sh
~/.../test/test\$

The screenshot shows a code editor with a file explorer on the left and a shell window on the right. The file explorer shows a directory structure with folders like 'foca', 'hacker-lab', 'scanner.sh', 'Proyecto', 'proyecto', 'test', and 'test'. The main editor window displays a Bash script titled 'linux3_scripting.md' with the following content:

```
#!/bin/bash

while true; do
    echo "1. Agregar tarea"
    echo "2. Listar tareas"
    echo "3. Eliminar tarea"
    echo "4. Salir"
    read -p "Selecciona una opción: " opcion

    case $opcion in
        1)
            read -p "Ingrese la tarea: " tarea
            echo "$tarea" >> tareas.txt
            ;;
        2)
            cat tareas.txt
            ;;
        3)
            # Implementar la función para eliminar tareas
            ;;
        4)
            ;;
        *)
            echo "Opción inválida"
            ;;
    esac
done
```

The shell window on the right shows the execution of the script, with the following output:

```
~/workspace$ #!/bin/bash

while true; do
    echo "1. Agregar tarea"
    echo "2. Listar tareas"
    echo "3. Eliminar tarea"
    echo "4. Salir"
    read -p "Selecciona una opción: " opcion

    case $opcion in
        1)
            read -p "Ingrese la tarea: " tarea
            echo "$tarea" >> tareas.txt
            ;;
        2)
            cat tareas.txt
            ;;
        3)
            # Implementar la función para eliminar tareas
            ;;
        4)
            ;;
        *)
            echo "Opción inválida"
            ;;
    esac
done
1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir
Selecciona una opción: 1
Ingrese la tarea: lavar la ropa
1. Agregar tarea
2. Listar tareas
3. Eliminar tarea
4. Salir
Selecciona una opción: 2
```

The screenshot shows a code editor with a file explorer on the left and a shell window on the right. The file explorer shows a directory structure with folders like 'foca', 'hacker-lab', 'scanner.sh', 'Proyecto', 'proyecto', 'test', and 'test'. The main editor window displays a Bash script titled 'linux3_scripting.md' with the following content:

```
#!/bin/bash

# Obtener la salida del comando y la sustituirá en su lugar. Por ejemplo, si haces
# for file in $(ls), el shell primero llamará a ls y luego iterará sobre esos valores.

Ejemplo

• Obtener la fecha actual y almacenarla en la variable "fecha"

fecha=$(date +%Y-%m-%d)
echo "La fecha actual es: $fecha"

• Obtener el número de archivos en el directorio actual y almacenarlo en la variable
"num_archivos"

num_archivos=$(ls -l | wc -l)
echo "Hay $num_archivos archivos en este directorio."

• Iterar sobre los archivos en el directorio actual

for file in $(ls); do
    echo "Procesando archivo: $file"
done
```

The shell window on the right shows the execution of the script, with the following output:

```
~/workspace$ for file in $(ls); do echo "Procesando archivo: $file"; done
2. Listar tareas
3. Eliminar tarea
4. Salir
Selecciona una opción: 4
Salido...
~/workspace$ fecha=$(date +%Y-%m-%d)
echo "La fecha actual es: $fecha"
La fecha actual es: 2025-10-08
~/workspace$ num_archivos=$(ls -l | wc -l)
echo "Hay $num_archivos archivos en este directorio."
Hay 26 archivos en este directorio.
~/workspace$ for file in $(ls); do
    echo "Procesando archivo: $file"
done
Procesando archivo: agenda.txt
Procesando archivo: backup_20251007.tar.gz
Procesando archivo: backup_log_20251007.txt
Procesando archivo: banco.txt
Procesando archivo: colores.txt
Procesando archivo: contactos.txt
Procesando archivo: estudiantes_por_id_desc.txt
Procesando archivo: estudiantes_por_nombre.txt
Procesando archivo: estudiantes_por_nota.txt
Procesando archivo: estudiantes.txt
Procesando archivo: foca
Procesando archivo: frutas.txt
Procesando archivo: hacker-lab
Procesando archivo: hello.txt
Procesando archivo: img
Procesando archivo: linux1_shell.md
Procesando archivo: linux2_commands.md
Procesando archivo: linux3_scripting.md
Procesando archivo: mcd.sh
Procesando archivo: mkcd.sh
Procesando archivo: proyecto
Procesando archivo: Proyecto
Procesando archivo: replit.nix
Procesando archivo: tareas.txt
Procesando archivo: test
~/workspace$
```

