

DESAFIO NUMERO 3

Consideraciones que se tuvieron en cuenta para proponer mejoras el script bash ([informe_del_sistema.sh](#)).

- 1) Agrupar la lógica del script en funciones y crear una función main() para que ejecute la lógica principal.
- 2) Mejorar el manejo de excepciones por ej si falla una función por algún parámetro detener el script
- 3) Mejorar los comentarios que están en el código

```
=====
#!/bin/bash

# Función para imprimir el encabezado
imprimir_encabezado() {
    echo "Informe del Sistema para: $(hostname)"
    echo ""
}

# Función para mostrar la fecha y hora
mostrar_fecha_hora() {
    echo "Fecha y Hora: $(date +%d/%m/%Y %H:%M:%S)"
    echo ""
}

# Función para mostrar el uso del disco
mostrar_uso_disco() {
    echo "Uso del disco del sistema (partición raíz /):"
    resultado=$(df -h | grep '^/')
    if [ -z "$resultado" ]; then
        echo "Error: No se pudo obtener el uso del disco."
        exit 1
    fi
    echo "$resultado"
    echo ""
}

# Función para mostrar usuarios logueados
mostrar_usuarios_logueados() {
    echo "Usuarios actualmente logueados:"
    resultado=$(who)
    if [ -z "$resultado" ]; then
        echo "Error: No se pudo obtener la lista de usuarios."
        exit 1
    fi
}
```

```
    echo "$resultado"
    echo ""
}
```

Función para mostrar el uso de memoria

```
mostrar_uso_memoria() {
    echo "Uso de memoria (en Megabytes):"
    resultado=$(free -m)
    if [ $? -ne 0 ]; then # Verifica el código de salida del último comando
        echo "Error: No se pudo obtener el uso de memoria."
        exit 1
    fi
    echo "$resultado"
    echo ""
}
```

Función para buscar procesos

```
buscar_proceso() {
    read -p "Ingresa el nombre o parte del nombre de un proceso: " proceso
    echo "Procesos que coinciden con '$proceso':"
    resultados=$(ps -aux | grep "$proceso") # Almacena la salida en una variable
    if [ -z "$resultados" ]; then # Verifica si la variable está vacía
        echo "Error: No se encontraron procesos que coincidan con '$proceso'."
        exit 1
    fi
    echo "$resultados" # Imprime los resultados si hay coincidencias
}
```

Función principal que ejecuta el informe

```
main() {
    imprimir_encabezado
    mostrar_fecha_hora
    mostrar_uso_disco
    mostrar_usuarios_logueados
    mostrar_uso_memoria
    buscar_proceso
}
```

Ejecutar la función principal

```
main
```

=====

```
o hernan@AhA:~/educacionit/devops-bootcamp/Desafios/Fase1$ ./informe_del_sistema.sh
Informe del Sistema para: AhA

Fecha y Hora: 11/12/2024 13:53:06

Uso del disco del sistema (partición raíz /):
/dev/sdb2      176G   25G  143G  15% /
/dev/sdd1      15G   40K   15G   1% /media/hernan/NUEVO VOL

Usuarios actualmente logueados:
hernan  seat0      2024-12-11 10:20 (login screen)
hernan  tty2        2024-12-11 10:20 (tty2)

Uso de memoria (en Megabytes):
          total      usado      libre  compartido  búf/caché  disponible
Mem:      7835      5829      467      381      2181      2006
Inter:    4095       51     4044

Ingresa el nombre o parte del nombre de un proceso: █
```

Explicación de las Funciones del Script:

1. Función para Imprimir el Encabezado

```
bash
imprimir_encabezado() {
    echo "Informe del Sistema para: $(hostname)"
    echo ""
}
```

- **Propósito:** Imprimir el encabezado del informe que incluye el nombre del host del sistema (hostname).
- **Comando:** `$(hostname)` : devuelve el nombre del sistema actual.

2. Función para Mostrar la Fecha y Hora

```
bash
mostrar_fecha_hora() {
    echo "Fecha y Hora: $(date +%d/%m/%Y %H:%M:%S)"
    echo ""
}
```

- **Propósito:** Imprimir la fecha y la hora actuales en un formato específico.
- **Comando:** `$(date +%d/%m/%Y %H:%M:%S)` obtiene la fecha y hora del sistema en formato día/mes/año hora:minuto:segundo.

3. Función para Mostrar el Uso del Disco

```
bash
mostrar_uso_disco() {
    echo "Uso del disco del sistema (partición raíz /):"
    resultado=$(df -h | grep '^/')
    if [ -z "$resultado" ]; then
        echo "Error: No se pudo obtener el uso del disco."
    fi
}
```

```
        exit 1
    fi
    echo "$resultado"
    echo ""
}
```

- **Propósito:** Mostrar el uso del disco de la partición raíz (/).
- **Comandos:**
 - `df -h`: Muestra información sobre el uso del disco en un formato legible.
 - `grep '^/':` Filtra la salida para mostrar solo la línea que representa la partición raíz.
- **Verificación:** Comprueba si el `resultado` está vacío. Si lo está, se muestra un error y se detiene el script.

4. Función para Mostrar Usuarios Logueados

```
bash
mostrar_usuarios_logueados() {
    echo "Usuarios actualmente logueados:"
    resultado=$(who)
    if [ -z "$resultado" ]; then
        echo "Error: No se pudo obtener la lista de usuarios."
        exit 1
    fi
    echo "$resultado"
    echo ""
}
```

- **Propósito:** Mostrar la lista de usuarios actualmente conectados al sistema.
- **Comando:** `who` devuelve la lista de usuarios conectados.
- **Verificación:** Similar a la función anterior, verifica si `resultado` está vacío para manejar posibles errores.

5. Función para Mostrar el Uso de Memoria

```
bash
mostrar_uso_memoria() {
    echo "Uso de memoria (en Megabytes):"
    resultado=$(free -m)
    if [ $? -ne 0 ]; then # Verifica el código de salida del último comando
        echo "Error: No se pudo obtener el uso de memoria."
        exit 1
    fi
    echo "$resultado"
    echo ""
}
```

- **Propósito:** Mostrar el uso de memoria del sistema en megabytes.
- **Comando:** `free -m` proporciona información sobre el uso de memoria en formato megabyte.
- **Verificación:** Comprueba el código de salida del comando `free -m`. Si hay un fallo, muestra un error y se detiene el script.

6. Función para Buscar Procesos

```
bash
buscar_proceso() {
    read -p "Ingresa el nombre o parte del nombre de un proceso: " proceso
    echo "Procesos que coinciden con '$proceso':"
    resultados=$(ps -aux | grep "$proceso") # Almacena la salida en una
variable
    if [ -z "$resultados" ]; then # Verifica si la variable está vacía
        echo "Error: No se encontraron procesos que coincidan con '$proceso'."
        exit 1
    fi
    echo "$resultados" # Imprime los resultados si hay coincidencias
}
```

- **Propósito:** Permitir al usuario buscar procesos en el sistema por nombre o parte del nombre.
- **Comandos:**
 - `read -p`: Solicita al usuario que ingrese el nombre del proceso a buscar.
 - `ps -aux | grep "$proceso"`: Muestra todos los procesos en ejecución y filtra los que coinciden con la cadena ingresada por el usuario.
- **Verificación:** Si no se encuentran procesos que coincidan, se informa al usuario de un error y se detiene el script.

Función Principal

```
bash
main() {
    imprimir_encabezado
    mostrar_fecha_hora
    mostrar_uso_disco
    mostrar_usuarios_logueados
    mostrar_uso_memoria
    buscar_proceso
}
```

- **Propósito:** Ejecutar todas las funciones definidas en el script en el orden correspondiente para generar el informe del sistema.

Ejecución del Script

```
bash
# Ejecutar la función principal
main
```

- Esta línea finalmente llama a la función `main`, comenzando así la ejecución del script y generando el informe del sistema.

Conclusiones

Este script utiliza varias construcciones de Bash para manejar errores de manera efectiva y proporciona información útil sobre el estado del sistema. Las verificaciones adecuadas aseguran que el script se detenga de manera segura si algo sale mal, lo que lo hace más confiable.