

Sistemas Operativos Primero:

Trabajo Integrador



Carrera: Tecnicatura Superior en Desarrollo de Software. Comision A.

Profesor: Taser Lucas.

Integrantes: Franco Andres Chocou, Aaron Serio Bravo, Jose Ignacio Gutierrez. Instituto Idra

Documentación Técnica

En el siguiente proyecto se realizó un script en bash el cual va a automatizar 3 tareas, es muy sencillo de utilizar e intuitivo por lo que no debería generar ningún problema al usuario a la hora de ejecutarlo.

El script va a automatizar lo siguiente:

- Generar un informe con el uso actual de la CPU, memoria y disco, y guardarlo en un archivo log.
- Verificar e instalar actualizaciones del sistema de forma automática, registrando los cambios.
- Eliminar archivos temporales, cache del sistema y de navegadores para liberar espacio en disco.

Los requisitos técnicos para poder ejecutar el script son:

1. Sistema operativo: Un sistema como Linux o MacOS. Windows permite ejecutarlo a través de WSL o Git Bash
2. Bash: Tener Bash instalado. Se puede comprobar previamente ejecutando `bash --version` en la terminal del sistema.

Las herramientas que necesitamos son un editor de Texto como por ejemplo:

1. Nano.
2. Vim.
3. Gedit.
4. Emacs.

Terminal: Un terminal o consola donde se pueda ejecutar los comandos y scripts

En cuanto a las dependencias, solo se necesita los comandos básicos de Unix: tales como `ls`, `cp`, `mv`, `rm`, etc.

Explicación detallada del código

En esta sección vamos a explicar paso por paso el código utilizado mostrando el código por partes y luego su explicación detallada:

1. Shebang

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

Esta línea indica que el script debe ser ejecutado usando el intérprete de Bash.

2. Definición de Colores

NC='\033[0m' # Sin color

RED='\033[0;31m' # Rojo

GREEN='\033[0;32m' # Verde

YELLOW='\033[0;33m' # Amarillo

BLUE='\033[0;34m' # Azul

Estas variables definen colores que se usarán en el texto del script. \033 es el código de escape para los colores en la terminal. NC se utiliza para restablecer el color al final de las líneas.

3. Función para Generar Informe de Uso de Recursos

```
generate_report()
{
    report_file="uso_recursos.log"
    echo -e "${BLUE}Generando informe de uso de recursos...${NC}"
    echo "Uso de CPU, Memoria y Disco" > "$report_file"
    echo "===== " >> "$report_file"
    echo -e "${YELLOW}Uso de CPU:${NC}" >> "$report_file"
    top -b -n1 | grep "Cpu(s)" >> "$report_file"
    echo "" >> "$report_file"
    echo -e "${YELLOW}Uso de Memoria:${NC}" >> "$report_file"
    free -h >> "$report_file"
    echo "" >> "$report_file"
    echo -e "${YELLOW}Uso de Disco:${NC}" >> "$report_file"
    df -h >> "$report_file"
    echo -e "${GREEN}Informe guardado en $report_file.${NC}"
}
```

Descripción: Esta función genera un informe sobre el uso de recursos y lo guarda en un archivo llamado uso_recursos.log.

echo -e: Permite interpretar los códigos de escape para colores.

top -b -n1 | grep "Cpu(s)": Obtiene el uso de CPU en modo batch (no interactivo) y filtra la salida para mostrar solo la línea con "Cpu(s)".

free -h: Muestra la memoria usada y libre de una manera legible.

df -h: Muestra el uso del disco en formato legible.

4. **Función para Verificar e Instalar Actualizaciones del Sistema**

```
install_updates()
{
    echo -e "${BLUE}Verificando e instalando actualizaciones del sistema...${NC}"
    sudo apt update &>> actualizaciones.log
    sudo apt upgrade -y &>> actualizaciones.log
    echo -e "${GREEN}Actualizaciones instaladas y registradas en
actualizaciones.log.${NC}"
}
```

Descripción: Verifica e instala actualizaciones del sistema usando apt (que es común en distribuciones basadas en Debian).

sudo apt update: Actualiza la lista de paquetes disponibles.

sudo apt upgrade -y: Instala las actualizaciones disponibles sin pedir confirmación.

&>> actualizaciones.log: Redirige tanto la salida estándar como los errores al archivo actualizaciones.log.

5. **Función para Eliminar Archivos Temporales y Caché**

```
clean_temp_files()
{
    echo -e "${BLUE}Eliminando archivos temporales y caché...${NC}"
    sudo apt-get clean
    rm -rf ~/.cache/*
    echo -e "${GREEN}Archivos temporales y caché eliminados.${NC}"
}
```

Descripción: Esta función limpia los archivos temporales y la caché del sistema.

sudo apt-get clean: Elimina archivos de paquetes descargados que ya no son necesarios.

rm -rf ~/.cache/*: Elimina todos los archivos en la carpeta de caché del usuario actual.

6. **Menú Interactivo**

```
while true; do
```

```
    echo -e "${BLUE}=== Menú Interactivo ===${NC}"
```

```
    echo "1. Generar informe de uso de recursos"
```

```
    echo "2. Verificar e instalar actualizaciones del sistema"
```

```
    echo "3. Eliminar archivos temporales y caché"
```

```
    echo "4. Salir"
```

```
    read -p "Selecciona una opción: " opcion
```

```
    case $opcion in
```

```
        1)
```

```
            generate_report;;
```

```
        2)
```

```
            install_updates;;
```

```
        3)
```

```
            clean_temp_files;;
```

```
        4)
```

```
            echo -e "${RED}Saliendo...${NC}"
```

```
            break;;
```

```
        *)
```

```
            echo -e "${RED}Opción no válida. Por favor, selecciona otra opción.${NC}";;
```

```
    esac
```

```
    echo # Línea en blanco para mejor legibilidad
```

```
done
```

Bucle while true: Este bucle permite que el menú se muestre repetidamente hasta que el usuario decida salir.

read -p: Muestra un mensaje y espera la entrada del usuario.

case \$opcion in: Estructura de control que evalúa la opción ingresada y llama a la función correspondiente.

break: Sale del bucle si el usuario selecciona la opción para salir.

echo: Utilizado para mejorar la legibilidad al agregar líneas en blanco.

Pruebas y validación

Menú general ingresando opción 1:

```
=== Menú de inicio ===
1. Generar informe de uso de recursos
2. Verificar e instalar actualizaciones del sistema
3. Eliminar archivos temporales y caché
4. Salir
Selecciona una opción: 1
Generando informe de uso de recursos...
Informe guardado en uso_recursos.log.
```

```
Uso de CPU, Memoria y Disco
=====
[ESC][0;33mUso de CPU:[ESC][0m
%Cpu(s): 60,0 us, 10,0 sy,  0,0 ni,  0,0 id,  0,0 wa,  0,0 hi, 30,0 si,  0,0 st

[ESC][0;33mUso de Memoria:[ESC][0m
              total      used      free      shared  buff/cache   available
Mem:          3,8Gi        414Mi       109Mi        17Mi        3,3Gi        3,1Gi
Swap:          1,0Gi         0,0Ki        1,0Gi

[ESC][0;33mUso de Disco:[ESC][0m
S.ficheros    Tamaño Usados  Disp Uso% Montado en
udev           1,9G      0  1,9G  0% /dev
tmpfs           394M    940K  393M  1% /run
/dev/sda1       19G     8,9G   8,8G 51% /
tmpfs           2,0G     12M   2,0G  1% /dev/shm
tmpfs           5,0M     4,0K   5,0M  1% /run/lock
tmpfs           394M     24K   394M  1% /run/user/1000
```

Menú general ingresando opción 2:

```
=== Menú de inicio ===
1. Generar informe de uso de recursos
2. Verificar e instalar actualizaciones del sistema
3. Eliminar archivos temporales y caché
4. Salir
Selecciona una opción: 2
Verificando e instalando actualizaciones del sistema...
Actualizaciones instaladas y registradas en actualizaciones.log.
```

```
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Obj:1 http://deb.debian.org/debian bullseye InRelease
Obj:2 http://deb.debian.org/debian-security bullseye-security InRelease
Obj:3 http://deb.debian.org/debian bullseye-updates InRelease
Obj:4 http://archive.raspberrypi.org/debian bullseye InRelease
Leyendo lista de paquetes...
Creando árbol de dependencias...
Leyendo la información de estado...
Se pueden actualizar 459 paquetes. Ejecute «apt list --upgradable» para verlos.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Leyendo lista de paquetes...
Creando árbol de dependencias...
Leyendo la información de estado...
Calculando la actualización...
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
 libopenpgl0 libre2-9 libwpe-1.0-1 libwpebackend-fdo-1.0-1 sse3-support
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
 gcc-10-base:amd64 krb5-locales libc6:amd64 libcom-err2:amd64 libcrvot1:amd64

20:13:44: Esto es Geany 1.37.1.
20:13:44: Archivo /home/aaron9406/hoja.txt abierto (1)
20:13:44: Estableciendo modo de sangría Espacios para /home/aaron9406/Desktop/ProyectoSO/proyecto.
20:13:44: Estableciendo modo de sangría Espacios para /home/aaron9406/Desktop/ProyectoSO/proyecto.
20:13:44: Archivo /home/aaron9406/Desktop/ProyectoSO/proyecto.sh abierto (2)
20:13:44: Estableciendo modo de sangría Espacios para /home/aaron9406/Desktop/ProyectoSO/uso_recur
do ancho de sangría a 2 para /home/aaron9406/Desktop/ProyectoSO/actualizaciones.log.
```

Menú general ingresando opción 3:

```
=== Menú de inicio ===
1. Generar informe de uso de recursos
2. Verificar e instalar actualizaciones del sistema
3. Eliminar archivos temporales y caché
4. Salir
Selecciona una opción: 3
Eliminando archivos temporales y caché...
Archivos temporales y caché eliminados.
```

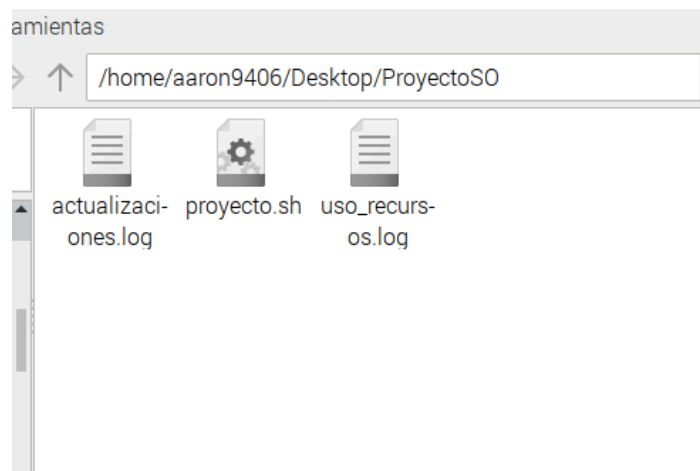
Opción incorrecta ingresada:

```
=== Menú ===  
1. Generar informe de uso de recursos  
2. Verificar e instalar actualizaciones del sistema  
3. Eliminar archivos temporales y caché  
4. Salir  
Selecciona una opción: 6  
Opción no válida. Por favor, selecciona otra opción.
```

Opción 4 Saliendo del programa:

```
=== Menú de inicio ===  
1. Generar informe de uso de recursos  
2. Verificar e instalar actualizaciones del sistema  
3. Eliminar archivos temporales y caché  
4. Salir  
Selecciona una opción: 4  
Saliendo...
```

Archivos Generales:



Reflexiones finales

Finalmente el trabajo se pudo realizar, tuvimos ciertas complicaciones a la hora de realizar el código, o de darle una mejor estética e instalar los paquetes principales para iniciar el proyecto.

Como posibles mejoras se podría implementar mas opciones al script, que muestre algo mas de información, además de mejorar la estética o la interfaz.