

Script Monitor_sistema.sh

En primer lugar, escribimos el sheban, esto lo que hace es indicar que el script va a ser ejecutado por la Shell /bash, es la primera línea que todo script de Linux suele tener a no ser que lo hagan en una Shell diferente.

Las 3 primeras líneas van a llamar para que se muestre en pantalla la información de usuario llamando a las variables \$USER y \$SHELL

```
GNU nano 7.2 monitor_sistema.sh
#!/bin/bash

echo "Información del usuario y el sistema"
echo "Nombre del usuario actual: $USER"
echo "Shell utilizada: $SHELL"

#Almacenamos la salida del comando en la variable InformacionSistemaOperativo
InformacionSistemaOperativo=$(hostnamectl | grep "Operating" | awk '{print $3 " " $4;}')

#llamamos a la variable
echo Sistema Operativo ejecutándose: $InformacionSistemaOperativo

read -p "Cual es su sistema Operativo preferido:" Sistema
echo $Sistema, "Ese sistema operativo es curioso"

#Redireccionamos el contenido de las variables al archivo sistema.txt

echo "Información del sistema:" > Sistema.txt
echo "Nombre del usuario actual:$USER" >> Sistema.txt
echo "Shell utilizada:$SHELL:$SHELL" >> Sistema.txt
echo "Sistema Operativo en ejecución:$InformacionSistemaOperativo">> Sistema.txt
echo "Utilización de la cpu:$InformacionCPU" >> Sistema.txt
echo "Utilización de memoria Ram:$InformacionMemoriaRam" >> Sistema.txt

#Utilizando variables locales,muestra en pantalla a continuación un mensaje
#estructurado que contenga información sobre el uso actual de CPU y de memoria
#RAM en porcentaje.

InformacionCPU=$(top -bn1 | grep "Cpu" | awk '{print ((100-$8 "%"))}')
echo $InformacionCPU, este es actualmente el uso de cpu.

InformacionMemoriaRam=$(top -bn1 | grep "MiB Mem" | awk '{print $8/$4 * 100 "%"}')
echo $InformacionMemoriaRam, Este es el porcentaje de memoria RAM ejecutándose.
```

Si abrimos el archivo con un sudo nano -l nos va a mostrar el número de cada línea, esto visualmente y a la hora de explicarlo es mucho mejor.

En la línea 9 se ejecuta el comando hostnamectl, comando que muestra todo tipo de información del sistema, lo que hacemos en el comando entero es filtrar con Grep el

nombre “Operating” y con el comando awk extraemos el tercer y cuarto campo los cuales corresponden al nombre y la versión del sistema.

```
tomathiko@tomathiko-server:~$ hostnamectl
Static hostname: tomathiko-server
Icon name: computer-vm
Chassis: vm
Machine ID: 3cdf0f403b1d4b5b8610f796d400751e
Boot ID: 8891571bcfab4f53b3d3478b73c1b288
Virtualization: vmware
Operating System: Ubuntu 24.04.1 LTS
Kernel: Linux 6.8.0-45-generic
Architecture: x86_64
Hardware Vendor: VMware, Inc.
Hardware Model: VMware Virtual Platform
Firmware Version: 6.00
Firmware Date: Thu 2020-11-12
Firmware Age: 3y 11month 1w 2d
tomathiko@tomathiko-server:~$
```

El resultado que nos da esto lo almacenamos en la variable `$InformacionSistemaOperativo`, para que el resultado se muestre en pantalla hacemos una llamada con `echo` Una breve frase que explique la variable, en este caso Sistema Operativo Ejecutándose: y llamamos a la variable que hemos declarado `$InformaciónSistemaOperativo`.

En la línea 14 realizamos un `read -p` donde escribimos una pregunta para que más tarde el usuario la complete y ese valor que ha escrito se guardara en la variable `sistema`. Después en la línea de abajo, imprimiremos el valor ingresado seguido de la frase “Ese es un sistema Operativo curioso”

De la línea 21-26

Redirigimos todas las variables que hemos declarado al archivo `Sistema.txt`

Estas líneas redirigen toda la información que hemos obtenida sobre el usuario (Shell, sistema operativo, CPU, memoria Ram, al archivo `sistema.txt`.

En la línea 21 se puede observar que escribimos únicamente un `>` y después `>>`, esto es debido a que si el archivo no existe lo crea, y si existe lo sobre escribe entero, entonces como nos interesa que solo sobrescriba una vez y se guarde toda la información de abajo después del primer archivo ponemos `>>` para que simplemente añada información al archivo sin sobrescribir.

En la línea 32 a la 34, vamos a desglosar este comando para que lo entendamos correctamente:

En esta línea lo que hacemos es ejecutar el comando `TOP -bn1` para obtener la información sobre la cpu que nos interesa. `-bn1` es = a `top -b, top -n, top -1`. Lo que le decimos al script es que vamos a ejecutar `top` en modo sin interfaz interactiva y solo se haga una iteración. Después filtramos con el comando `grep “Cpu”` para obtener la información de la cpu y utilizamos el `awk ‘{print (100-$8)}’` aquí lo que hacemos es restar el valor de la columna \$8 que representa el porcentaje de la cpu que están en inactividad y los restamos al 100% para obtener el porcentaje de uso de la cpu.

Esta información la guardamos en la variable `InformaciónCPU`.

En segundo lugar, en la línea 35 a las 36, utilizamos el mismo método, vamos a ejecutar el comando top con la misma iteración, pero para obtener el uso de la memoria ram, filtramos por “MiB Mem” y con awk calculamos el porcentaje de memoria utilizada del sistema dividiendo el valor de \$8 (la memoria utilizada) entre \$4 la memoria total disponible y esto lo multiplicamos por 100 para obtener el porcentaje de uso de la memoria RAM

Por último, en las dos últimas líneas donde hemos declarado las variables de \$InformaciónCPU y \$InformacionMemoriaRam con un echo hacemos que aparezcan en pantalla la información declara. Una vez terminado el script realizamos un:

Sudo chmod +x para darle permisos de ejecución.

```
tomathiko@tomathiko-server:~$ ls -l
total 80
drwxrwxr-x 3 tomathiko tomathiko 4096 oct  8 11:08 Desktop
drwxrwxr-x 3 tomathiko tomathiko 4096 oct 14 09:08 Documents
drwxrwxr-x 2 tomathiko tomathiko 4096 oct  8 11:06 Downloads
-rw-rw-r-- 1 tomathiko tomathiko  17 oct 14 09:25 ficherosseguro.txt
-rw-rw-r-- 1 tomathiko tomathiko 104 oct 14 10:18 ficherosseguro.txt.gpg
drwxrwxr-x 2 tomathiko tomathiko 4096 oct  8 11:06 imagenes
-rwxrwxr-x 1 tomathiko tomathiko 766 oct  9 09:38 info_sistema.sh
-rwxrwxr-x 1 tomathiko tomathiko 1417 oct 22 07:30 monitor_sistema.sh
-rwxrwxr-x 1 tomathiko tomathiko 497 oct 16 10:12 operaciones1.sh
-rw-rw-r-- 1 tomathiko tomathiko 798 oct 15 12:23 parpos2.sh
-rw-rw-r-- 1 tomathiko tomathiko 226 oct 15 11:14 parpos.sh
-rw-r--r-- 1 root      root      94 oct 14 09:13 salida.txt
-rwxrwxr-x 1 tomathiko tomathiko 106 oct 16 09:14 script1.sh
```

Ahora ejecutamos el Script.

```
tomathiko@tomathiko-server:~$ ./monitor_sistema.sh
Información del usuario y el sistema
Nombre del usuario actual: tomathiko
Shell utilizada: /bin/bash
Sistema Operativo ejecutándose: Ubuntu 24.04.1
Cual es su sistema Operativo preferido:Ubuntu
Ubuntu, Ese sistema operativo es curioso
6%, este es actualmente el uso de cpu.
13.6505%, Este es el porcentaje de memoria RAM ejecutándose.
tomathiko@tomathiko-server:~$
```

Y hacemos un CAT del archivo del sistema.txt y nos debería salir toda la información

```
tomathiko@tomathiko-server:~$ cat Sistema.txt
Información del sistema:
Nombre del usuario actual:tomathiko
Shell utilizada::/bin/bash
Sistema Operativo en ejecución:Ubuntu 24.04.1
Utilización de la cpu:
Utilización de memoria Ram:
tomathiko@tomathiko-server:~$
```

Posibles Errores: En el cat Sistema.txt es posible que no me muestre la información del porcentaje utilizado de la cpu y de la ram debido a que es probable que con el comando que hemos ejecutado en el script la información de la memoria ram y del cpu se esta calculando después de intentar redirigirlas al archivo sistema.txt