

Introducción a Linux

Módulo 5



Redireccionamiento

Tipos de redirecciones

Hay tres tipos de redirecciones a tener en cuenta:

- STDIN (0) -> Entrada Estándar.
- STDOUT (1) -> Salida Estándar.
- STDERR (2) -> Salida Estándar de error.

A continuación, las veremos en detalle.

Salida Estándar (STDOUT)

La salida estándar es la salida normal de un comando. De manera predeterminada esta salida se muestra en la terminal del usuario.

```
[rino@restauracion scripts]$ ls
test          educacionit  sakis3g
117-101.pdf   file         scripts
117-101.txt   filescripts  ssh2
agregarurl    filescripts~ test.html
```

Al ejecutar el comando **ls** el resultado se muestra en pantalla.

La salida de un comando se puede redireccionar a un archivo con la ayuda del signo mayor ">" como se ve en el siguiente ejemplo:

```
[rino@restauracion scripts]$ ls > salida.txt
```

Esto generará un archivo llamado **salida.txt**, que contendrá el resultado del comando **ls**. Para verificarlo se puede utilizar el comando **cat salida.txt**

Si el archivo **salida.txt** existiera, el contenido será sobrescrito con el resultado de la redirección.

Para poder agregar contenido al final del archivo, sin sobrescribir el contenido existente, se puede utilizar doble signo mayor ">>". Ejemplo:

```
[code]
[rino@restauracion scripts]$ ls >> salida.txt
[rino@restauracion scripts]$ ls -la salida.txt
-rw-rw-r-- 1 rino rino 580 Sep 27 21:38 salida
[rino@restauracion scripts]$ ls >> salida.txt
[rino@restauracion scripts]$ ls -la salida.txt
-rw-rw-r-- 1 rino rino 870 Sep 27 21:39 salida
[/code]
```

Como se puede ver el archivo incrementó su tamaño, esto se debe a que ejecutamos dos veces el comando **ls** con >> lo cual agregó el resultado del **ls**, manteniendo el contenido del primer **ls**.

Salida Error (STDERR)

La salida de estándar de Error, es la salida de un comando que no terminó de manera satisfactoria.

```
[code]
[rino@restauracion testing]$ ls -z
ls: invalid option -- z
Try `ls --help' for more information.
[/code]
```

Al ejecutar el comando **ls** con una opción incorrecta, el error se muestra en pantalla de manera normal. En este caso va a cambiar el tratamiento para poder redirigir el resultado a un archivo.

La salida de error de un comando se puede redireccionar a un archivo con un *dos* y signo mayor “**2>**”, como se ve en el siguiente ejemplo:

```
[rino@restauracion testing]$ ls -z 2> error.txt
```

Al igual que con la salida estándar, en caso de que el archivo **error.txt** no exista lo creará, y en caso de existir lo sobrescribirá.

Para mantener el contenido sin sobrescribirlo se utiliza “**2>>**”.

Con esto se puede diferenciar en distintos archivos ambas salidas de un mismo comando:

```
[rino@restauracion testing]$ ls  
archivoquenoexiste * 1> salida.txt 2>  
error.txt
```

En este ejemplo el error mostrado por el archivo que no existe será guardado en **error.txt** mientras que el resultado del asterisco se guardará en **salida.txt**.

Otra opción es redireccionar ambas salidas a un mismo archivo:

```
[rino@restauracion testing]$ ls  
archivoquenoexiste * > ambos.txt 2>&1
```

En este caso, lo que se hace es redireccionar la salida a **ambos.txt**. Luego, con el **2>&1** se está redirigiendo el resultado del **STDERR** (que tiene asociado el número 2) al **STDOUT** (que tiene asociado el número 1). **Es importante recordar el orden en el que se dirige el contenido.** Al final se agrega la redirección de **STDERR** a **STDOUT**.

Entrada Estándar (STDIN)

La entrada estándar se utiliza para tomar el contenido de un archivo a un comando.

En el ejemplo de la derecha, hay un archivo que se llama **desordenado.txt**, que contiene números desordenados.

[code]

```
# echo "2" > orden.txt  
# echo "3" >> orden.txt  
# echo "5" >> orden.txt  
# echo "1" >> orden.txt
```

```
[crond1@oc6127656113 ~]$ cat orden
```

```
2
```

```
3
```

```
5
```

```
1
```

```
[/code]
```

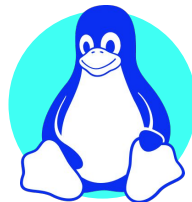
Al ejecutar el comando **sort < desordenado.txt** pasa el contenido del archivo **desordenado.txt** al comando **sort**, el cual mostrará el resultado a través del **STDOUT**.

```
[code]
[crond1@oc6127656113 ~]$ sort < desordenado.txt
1
2
3
5
[/code]
```

También se puede redirigir la salida a un archivo:

```
[crond1@oc6127656113 ~]$ sort <
desordenado.txt > ordenado.txt
```

De esta manera, el resultado ordenado por el comando **sort** se guarda en el archivo **ordenado.txt**



**¡Sigamos
trabajando!**