

# Redirecciones y tuberías en Bash Linux

# Introducción: entradas y salidas

La **entrada estándar(stdin)** representa los datos que necesita una aplicación para funcionar, como por ejemplo un archivo de datos o información ingresada desde la terminal y es representado en la terminal como el tipo 0.

La **salida estándar (stdout)** es la vía que utilizan las aplicaciones para mostrarte información, allí podemos ver el progreso o simplemente los mensajes que la aplicación quiera darte en determinado momento y es representado en la terminal como el tipo 1.

El **error estándar (stderr)** es la forma en que los programas te informan sobre los problemas que pueden encontrarse al momento de la ejecución y es representado en la terminal como el tipo 2.

Todos estos tipos son representados físicamente como archivos en el sistema, pues como ya debes saber, **todo en Linux son archivos**.

# Redirecciones >

Una redirección consiste en trasladar la información de un tipo a otro, por ejemplo de la salida estándar a la entrada estándar o del error estándar a la salida estándar. Esto lo logramos usando el símbolo >.

Por ejemplo, para redireccionar la salida de un comando y volcarla a un archivo bastaría con ejecutar:

**> *ls -la > archivo.txt***

Resuminedo: Redireccionamiento de salida: Con él podremos enviar la salida de un comando hacia un fichero, otro comando o un dispositivo

# Redireccion >>

Si usamos la redirección > en el comando:

```
> ls -la > archivo.txt
```

cada vez que ejecutemos este comando el contenido de archivo.txt será reemplazado por la salida del comando ls.

Si queremos **agregar** la salida del comando al archivo, en lugar de reemplazarla, entonces ejecutamos el siguiente comando usando >>:

```
> ls -la >> archivo.txt
```

**Resumiendo:** Redireccionamiento de adición: Es un redireccionamiento de salida con la peculiaridad de que la salida la añade al destino indicado, normalmente un fichero, y por tanto no borra su contenido.

*Ángel González M.*

# Redirección del error

Lo interesante es que, además de la salida estándar, también podemos redireccionar el error estándar y la entrada estándar. Si queremos forzar a que un programa nos imprima en pantalla los errores que consiga durante su ejecución podemos redireccionar el error estándar hacia la salida estándar.

Eso lo logramos ejecutando:

**> *programa 2> logErrors.txt***

En le ejemplo anterior guardamos los mensajes solo cuando programa da error.

Resumiendo: Redireccionamiento de errores: De esta manera sólo redirigimos los errores que produzca el comando hacia otro dispositivo de salida que no sea la pantalla. Puede ser de adición también.

```
>cp doc1 doc2 2> /dev/null
```

```
>cp doc1 doc2 2>> fichero_error.log
```

Se pueden combinar la salida estándar con la de errores.

```
>cp -v /boot/* /root/copia/ >> boot.log 2>> error.log
```

# Redirección &> &>>

Redireccionamiento completo: De esta manera redirigimos con un solo comando tanto la salida normal como la salida de errores al mismo destino, puede ser de adición también. Los siguientes ejemplos son similares en su resultado.

```
cp -v /boot/* /root/copia/ >> boot.log 2>> boot.log
```

```
cp -v /boot/* /root/copia/ &>> boot.log
```

# Redireccion <

Podemos, por ejemplo, contar las líneas que tiene un archivo redireccionando la entrada estándar de `wc` hacia un archivo de texto. Así:

```
> wc < archivo.txt
```



&1 &2 &3 ...

&1 &2 &3 ...

Redireccionamiento por parámetros: De esta manera redirigimos a un destino que ya se ha nombrado anteriormente en el mismo comando, el número indica la posición de ese destino. Vemos como el ejemplo anterior

```
cp -v /boot/* /root/copia/ >> boot.log 2>> boot.log
```

también se podría escribir de la siguiente manera.

```
cp -v /boot/* /root/copia/ >> boot.log 2>> &1
```

# /dev/null

También podemos hacer algo muy común en la administración de sistemas, descartar el error estándar de un proceso. Para eso ejecutamos:

```
> programa 2> /dev/null
```

O incluso descartar su salida estándar:

```
> programa > /dev/null
```

En Linux, /dev/null es un archivo especial al que se envía cualquier información que quiera ser descartada.

# Tuberías

Uno de los principios de la filosofía Unix consiste en tener aplicaciones muy pequeñas que hagan tareas muy puntuales y que usadas en conjunto puedan realizar tareas complejas.

Entonces, siguiendo este principio debe existir una forma para que varias aplicaciones puedan interactuar entre sí, aquí es donde entran las tuberías.

# Tuberías: introducción

Una tubería en Linux no es más que una forma práctica de redireccionar la salida estándar de un programa hacia la entrada estándar de otro. Esto se logra usando el símbolo | (pipe).

Resumiendo: De esta forma podemos ejecutar un comando y hacer que el resultado de éste sea la entrada para otro, que a su vez puede servir de entrada a un tercero.

Por ejemplo, podríamos ver los procesos que están corriendo en el sistema usando ps y le redireccionamos la salida a sort para que los ordene por PID:

> ***ps -a | sort***

También podemos redireccionar la salida estándar del comando cat y pasarla como entrada estándar del comando wc para contar las líneas y palabras de un archivo:

> ***cat archivo.txt | wc***

# Más ejemplos

Lista los archivos y directorios pero ordenados de la Z a la A

```
>ls | sort -r
```

Lista los archivos y directorios de la ruta /usr/bin pero con paginación(poco a poco)

```
>ls /usr/bin | less
```

lista los archivos y directorios pero ordenados de la Z a la A y se queda solo con la primera linea

```
>ls | sort -r | head -1
```

# Más ejemplos

Ordena (alfabéticamente) el fichero Agenda.txt y lo guarda como Agenda.ord

```
>sort < Agenda.txt > Agenda.ord
```

la lista de usuarios ordenada alfabéticamente, aparece por pantalla y se almacena en el fichero listin

```
>who | sort | tee listin | more
```

# Más ejemplos

Busca un proceso (programa) que se esté ejecutando de nombre nemo

```
>ps -e | grep nemo
```

Muestras cuantos usuarios hay que su nombre contenga an

```
>cat /etc/passwd | grep an | wc -l
```

Muestras cuantos usuarios hay que su nombre comience por an

```
>cat /etc/passwd | grep ^an | wc -l
```