

REDIRECCIONAMIENTO Y PIPELINES

DESARROLLO

3.1 Generalidades de la salida estándar

Cuando empieza a pasar el tiempo y más utilizamos los comandos en el sistema GNU/Linux más empezamos a dominar sus salidas.

Esto permite que la ejecución de un comando no se visualice inmediatamente en pantalla, sino que su salida sea la entrada de otro comando o fichero, permitiendo que podamos ejecutar varias órdenes en una sola línea de comando.

Cuando ingresamos una orden al sistema se conoce como **stdin**, cuando visualizamos una respuesta lo conocemos como **stdout** o una respuesta de error **stderr**.

Cuando se inicia una sesión se habilitan tres procesos que corresponden a stdin, stdout y stderr los cuales son enumerados de la siguiente así:

- El proceso 0 es la entrada estándar (stdin) y corresponde a lo digitado por el teclado.
- El proceso 1 es la salida estándar (stdout) y corresponde a lo mostrado en la pantalla.
- El proceso 2 es la salida de errores (stderr) y corresponde a los mensajes de error visualizados en pantalla. (Obasoft.es, 2017).

Se debe tener presente que al escribir un comando en consola para ser ejecutado por lo general esta proviene del teclado y la salida estándar o salida de error son vistos en pantalla pero los dos son ficheros totalmente independientes. De esta manera podemos controlar que la salida de un proceso sea la entrada de otro y que a partir de una entrada se pueda modificar la salida.

3.2 Redireccionamiento y pipelines

Cuando se redirrecciona lo que se hace es que la información que debería salir en la pantalla se “redirrecciona” valga la redundancia a otro sitio, en este caso un fichero.

Existen diferentes maneras, las más populares son:

- **>: Comando > Archivo**
Redirrecciona la salida de la ejecución hacia un archivo existente, si el archivo no llegase a existir este comando lo creará, pero si el archivo existe se modifica su contenido.
- **>>: Comando >> Archivo**
Redirrecciona y escribirá en el archivo file. Si el archivo ya existe la lista se escribirá al final del archivo, manteniendo la información previa. Si no existiese lo creará.
- **<: Comando < Archivo**
A entrada se lee desde el archivo, para este caso el fichero de donde se leyó la información no sufrirá ningún tipo de modificación.

En el siguiente ejemplo vamos a ejecutar el comando `ls -l` que nos va a listar el contenido del directorio de trabajo del usuario unicor en el cual me estoy y redirrecciona la salida almacenando la información en el archivo de nombre MisFicheros, luego si queremos ver su contenido simplemente ejecutamos el comando `cat`:

unicor@ubuntu:~\$ ls -l muestra el contenido del directorio.

unicor@ubuntu:~\$ ls -l >MisFicheros redirreciona a la salida y la guarda en un archivo que se llama MisFicheros.

unicor@ubuntu:~\$ cat MisFicheros Muestra el contenido y allí aparece también el nuevo archivo creado.

```
unicor@ubuntu:~$ ls -l
total 52
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Descargas
drwxr-xr-x 2 uncor uncor 4096 Jul 11 19:36 Documentos
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Escritorio
drwxrwxr-x 5 uncor uncor 4096 Jul 11 19:32 examenes
-rw-r--r-- 1 uncor uncor 8980 Jul 10 17:39 examples.desktop
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Imágenes
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Música
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Plantillas
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Público
drwxrwxr-x 2 uncor uncor 4096 Jul 11 19:23 trabajos
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Vídeos
unicor@ubuntu:~$ ls -l >MisFicheros
unicor@ubuntu:~$ cat MisFicheros
total 52
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Descargas
drwxr-xr-x 2 uncor uncor 4096 Jul 11 19:36 Documentos
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Escritorio
drwxrwxr-x 5 uncor uncor 4096 Jul 11 19:32 examenes
-rw-r--r-- 1 uncor uncor 8980 Jul 10 17:39 examples.desktop
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Imágenes
-rw-rw-r-- 1 uncor uncor 0 Jul 17 15:04 MisFicheros
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Música
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Plantillas
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Público
drwxrwxr-x 2 uncor uncor 4096 Jul 11 19:23 trabajos
drwxr-xr-x 2 uncor uncor 4096 Jul 11 17:25 Vídeos
unicor@ubuntu:~$
```

Figura 1: Ejemplo de utilización del comando ls -l

Fuente: Elaboración propia

También se escribiría una cadena de texto la cual se almacenara en un fichero llamado ingSistemas, como el fichero no existe este se crea en el directorio en el que me encuentro:

unicor@ubuntu:~\$ echo "Ingeniería de Sistemas Linea Uno" > ingSistemas

El comando echo muestra una cadena de texto, para este caso esto se almacenará en el fichero ingSistemas. Entonces el contenido del fichero será:

```
unicor@ubuntu:~$ echo "Ingeniería de Sistemas Linea Uno" > ingSistemas
unicor@ubuntu:~$ cat ingSistemas
Ingeniería de Sistemas Linea Uno
unicor@ubuntu:~$
```

Figura 2: Figura 3: Ejemplo de utilización del comando ls -l

Fuente: Elaboración propia

Ahora si quiero anexar una nueva línea al fichero ingSistemas sin perder la información ya almacenada utilizo el siguiente comando:

```
unicor@ubuntu:~$ echo "Ingeniería de Sistemas Linea Dos" >> ingSistemas
```

Entonces el contenido del fichero será la información anterior más la nueva información que se anexo o añadió, sin que se perdiera la información que existía en el fichero:

```
unicor@ubuntu:~$ echo "Ingeniería de Sistemas Linea dos" >> ingSistemas
unicor@ubuntu:~$ cat ingSistemas
Ingeniería de Sistemas Linea Uno
Ingeniería de Sistemas Linea dos
unicor@ubuntu:~$
```

Figura 4: Ejemplo de redireccionamiento

Fuente: Elaboración propia

Pipelines

Los pipelines son conocidos como tuberías, estos hacen que podamos concatenar varias peticiones en una sola línea, la entrada de una orden es la salida de la orden anterior. El símbolo utilizado para representar el pipeline es el "|".

Ejemplos utilizando tuberías:

Para guardar la información de los grupos correspondiente a algún usuario específico del sistema en un fichero, podemos usar el siguiente comando:

```
unicor@ubuntu:~$ cat /etc/group | grep "unicor" | tee grupoUnicor
```

```
unicor@ubuntu:~$ cat /etc/group | grep "unicor" | tee grupoUnicor
adm:x:4:syslog,unicor
cdrom:x:24:unicor
sudo:x:27:unicor
dip:x:30:unicor
plugdev:x:46:unicor
lpadmin:x:113:unicor
unicor:x:1000:
smbashare:x:128:unicor
unicor@ubuntu:~$
```

Figura 5: Ejemplo utilizando tuberías

Fuente: Elaboración propia

El comando `cat /etc/group` despliega la información de los grupos creados en el sistema, luego el comando `grep` busca todas las líneas del fichero donde aparece la palabra **Unicor** el cual es el nombre del usuario y por ultimo con el comando `tee` guardamos la información desplegada en el fichero `gupoUnicor`. Si queremos ver la información de dicho fichero ejecutamos el comando:

```
unicor@ubuntu:~$ cat grupoUnicor
```

Si un usuario desea mostrar el contenido un determinado fichero, que contiene la ruta de los ficheros con extensión `.odt` en su directorio de trabajo, puede utilizar el siguiente comando:

```
unicor@ubuntu:~$ find /home/unicor -name "*.odt" | tee salidaODT | cat
```

El comando `find /home/unicor -name "*.odt"` busca todos los ficheros con extensión `.odt` existentes dentro del directorio `/home/unicor` o cualquiera de los sub-directorios, luego la información de salida se almacena en el fichero `salidaODT` con el comando `tee salidaODT` por ultimo el comando `cat` muestra en pantalla la ruta de estos ficheros ejemplo:

```
unicor@ubuntu:~$ find /home/unicor/Documentos/ -name "*.odt" | tee salidaODT | cat  
at  
/home/unicor/Documentos/carta1.odt  
/home/unicor/Documentos/documentopr.odt  
unicor@ubuntu:~$
```

Figura 6: Ejemplo utilización tuberías

Fuente: Elaboración propia

EJERCICIOS/ACTIVIDADES
<p>Actividad en clase 1. Realizar un taller utilizando la consola donde se evidencia la aplicación del redireccionamiento y las tuberías con los comandos >, >>, </p> <p>Metodología Cada estudiante va a realizar un taller donde documente varios ejemplos prácticos de redireccionamiento utilizando >, >> y </p>

BIBLIOGRAFIA
<p>OBASOFT.ES. (2017). Redireccionamientos el sistema de archivos en Linux. Recuperado de: http://www.obasoft.es/CF/SIINF/SIINF_04_Contenidos/redireccionamientos.html</p>