

## CAPÍTULO 2: EL SHELL DEL SISTEMA

- 2.1. La orden `tee` es un filtro especial que recoge los datos de la entrada estándar y los redirige a la salida normal y a un fichero especificado, tanto en operaciones de escritura como de anexo. Esta es una orden muy útil que suele usarse en procesos largos para observar y registrar la evolución de los resultados. El siguiente ejemplo muestra y registra el proceso de compilación e instalación de una aplicación GNU.

```
configure 2>&1 | tee aplic.sal  
make      2>&1 | tee -a aplic.sal  
make install 2>&1 | tee -a aplic.sal
```

Se propone como ejercicio interpretar la siguiente orden:

```
$ ls | tee salida | sort -r
```

Tal y como se indica la orden `tee` “duplica” los datos de la entrada, tanto en la salida estándar como en un fichero. Por lo tanto, mediante `ls | tee salida` se está haciendo un `ls` al directorio local y la salida de este proceso se muestra en la salida estándar y se almacena a su vez en el fichero `salida`. Con la parte final (`sort -r`) se ordena en modo alfabéticamente inverso la salida del comando `ls`, no así el contenido del fichero que mantendrá la salida de `ls` ordenado alfabéticamente.

- 2.2. Suponga que tiene permiso para escribir en un fichero, pero no para eliminarlo. Indique qué comando puede utilizar para vaciar el contenido del fichero sin utilizar ningún editor de textos.

Se puede utilizar `/dev/null` para obtener una cadena vacía con la que sobrescribir el fichero:

```
cat /dev/null > fichero
```

De este modo no se elimina el fichero, ya que no se tiene permiso, pero se lleva a cabo un borrado del contenido del mismo.

- 2.3. Genere un fichero llamado `respuesta.txt` con algún contenido y

**posteriormente realice las siguientes actividades.**

```
$ > respuesta.out < respuesta.txt cat
```

- ¿Qué realiza el comando anterior?

Se utiliza la redirección para almacenar en el fichero *respuesta.out* el contenido del fichero *respuesta*, el cual se lee mediante la orden *cat*, con lo que se consigue copiar el contenido de *respuesta* en *respuesta.out*

- ¿Cuál sería el modo convencional de realizarlo?

Como se ha visto, lo anterior sencillamente realiza una copia del fichero, para lo cual se puede utilizar la orden *cp*:

```
$ cp respuesta respuesta.out
```

- 2.4. Un usuario desea buscar en los ficheros *a*, *b* y *c* la palabra *back*, redirigiendo el resultado al fichero *a.out*. Para ello utiliza el comando *grep*, pero por equivocación no introduce el punto (.) en el nombre del fichero de salida, quedando la orden como sigue.**

```
$ grep back a b c > a out
```

¿Qué ha sucedido? Si recuerda, lo primero que lleva a cabo la redirección es vaciar el contenido del fichero de destino si éste existe, tal y como es el caso, ya que debido al error tipográfico el fichero de destino de la búsqueda es el fichero *a*, el cual también es origen de la misma. Tras el proceso se habrá perdido el contenido original del fichero *a*.

- 2.5. Suponga que tiene un fichero de texto con los siguientes datos de usuario: *Nombre*, *Apellido1*, *Apellido2*, *Año de nacimiento* y *ciudad de nacimiento*. Cada usuario aparece en una línea independiente y los datos están separados por tabuladores. Un ejemplo puede ser el siguiente:**

|             |           |           |         |         |
|-------------|-----------|-----------|---------|---------|
| Gonzalo     | Puga      | Sabio     | 1982    | Granada |
| Julio Gomez | Lopez     | 1975      | Murcia  |         |
| Jose Perez  | Perez     | 1958      | Almeria |         |
| Magdalena   | Fernandez | Fernandez | 1980    | Sevilla |

Se desea obtener una salida un poco más clara, ya que sólo se necesita el *Nombre*, *Apellido1* y *Año de Nacimiento*. Cree para ello un script con *awk* que devuelva el siguiente resultado.

| Nombre    | Apellido1 | Fecha<br>Nacimiento |
|-----------|-----------|---------------------|
| Gonzalo   | Puga      | 1982                |
| Julio     | Gomez     | 1975                |
| Jose      | Perez     | 1958                |
| Magdalena | Fernandez | 1980                |

La utilidad *awk* permite utilizar un fichero como fuente del formato a aplicar. Para indicar este extremo se utiliza el modificador *-f*. Por lo tanto, tan sólo habrá que crear un fichero que muestre la cabecera en el apartado *BEGIN* y que posteriormente imprima de modo tabular el contenido de las columnas deseadas. Dicho fichero puede ser el siguiente:

```
BEGIN {
    print "
    print "Nombre      Apellido1      Fecha"
    print "-----"
}
{
    printf "%-15s %-25s %2d\n", $1, $2, $4
}
```

Ya sólo quedaría indicar a *awk* que lo utilice para dar formato al fichero de datos:

```
$ awk -f ficheroFormato ficheroDatos
```