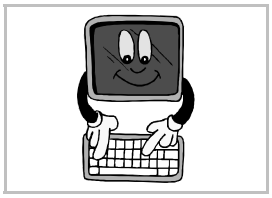

Trabajo en consola

Filosofía de diseño

Uno de los objetivos de UNIX es crear pequeños programas que realicen rápida y eficazmente su función y que se integren fácilmente entre sí. Para resolver problemas más complejos, se usan órdenes que van conectando el resultado de un programa con la entrada de otro. En vez de desarrollar grandes programas, complejos de mantener, se requiere un poco de ingenio por parte del usuario para que éste descubra cómo conectar los pequeños programas que tiene a su disposición para llegar a resolver su problema.

En esta hoja se van a explicar algunos de estos programas, llamados genéricamente **filtros**, así como la manera de conectarlos. Se recomienda leer la información *man* e *info* de todos los programas citados y seguir las instrucciones delante del teclado.

Origen

Esta filosofía fue enunciada por primera vez en el libro *Software Tools* de Brian Kernighan y P. J. Plauger, editado en 1976.

Entrada y salida estándar

Estos pequeños programas UNIX, los filtros, toman sus datos de la entrada estándar y dan sus resultados en la salida estándar. En principio, la entrada estándar es el teclado y la salida estándar es la pantalla, pero se pueden redireccionar para que sean un archivo, un dispositivo (la impresora, por ejemplo) u otro programa.

Ejemplo

El programa *sort* clasifica líneas de texto por orden alfabético. Si se le pasan las líneas “Uno”, “Dos” y “Tres”, debe devolver las líneas “Dos”, “Tres” y “Uno”. En la ilustración de la derecha se ve el proceso, junto con las teclas que hay que pulsar. El Ctrl D es necesario para indicar el fin de los datos que se pasan a *sort*. Obsérvese cómo *sort* ha tomado los datos que se han introducido por el teclado y ha emitido el resultado por la pantalla.

Entrada	Salida
Sort↵ **Uno**↵ **Dos**↵ **Tres**↵ Ctrl D	**Dos** **Tres** **Uno**

Redirección de la salida

El carácter ‘>’ sirve para indicar dónde hay que dirigir la salida del proceso. Si se dirige a un archivo y éste ya existía, se pierde su contenido original.

Ejemplo

El programa *cat* reproduce por la salida estándar todo lo que recibe por la entrada. Normalmente se usa para visualizar en pantalla un archivo. En este ejemplo se le van a dar los datos por el teclado y se van a redireccionar a un archivo. A la derecha se ve lo que hay que teclear. Se introducen por el teclado tres nombres y *cat*, en vez de mostrarlos por pantalla, los vuelca en el archivo **nombres**. Nota: conviene comprobar que efectivamente se ha creado correctamente el archivo, tecleando **`cat nombres`**.

Entrada
cat >nombres↵ **Carlos**↵ **Roberto**↵ **Adela**↵ Ctrl D

Redirección de la entrada

El carácter ‘<’ sirve para indicar de dónde hay que tomar la entrada estándar.

Ejemplo

Se le pueden pasar a *sort* los nombres que se acaban de almacenar en el archivo **nombres** y debe devolverlos por pantalla ordenados. A la derecha se ve lo que hay que escribir y la salida esperada.

Entrada	Salida
sort <nombres↵	**Adela** **Carlos** **Roberto**

Redirección de las dos

Por supuesto, se pueden redirigir simultáneamente tanto la entrada como la salida, usando los dos símbolos que se han explicado.

Ejemplo

El resultado del *sort* del ejemplo anterior se podría redirigir a un archivo, para guardar la lista ordenada de los nombres. Así: **`sort <nombres >ordenado`**. Esto crea el archivo `ordenado` (compruébese con **`cat ordenado`**).

Redirección no destructiva

Si se usa la construcción '>>' en vez de '>' para redirigir la salida de un programa a un archivo, el archivo original no se destruye, sino que el resultado del programa se añade al contenido del archivo.

Ejemplo

La orden *echo* envía a la salida estándar una línea de texto. La orden mostrada a continuación añadirá el nombre "Beatriz" al archivo `nombres`: **`echo Beatriz >> nombres`**. Compruébese que ahora el archivo `nombres` tiene cuatro líneas.

Tuberías

En inglés se denominan *pipes*. Se forma una tubería cuando se conecta la salida de un archivo con la entrada del siguiente. Se representa con el símbolo '|'.

Ejemplo

Para ver el contenido del archivo nombres clasificado por orden alfabético basta esta orden:

`cat nombres | sort`

Explicación: *cat* envía a la salida estándar las cuatro líneas del archivo `nombres`; por la tubería, esas cuatro líneas se dirigen a la entrada estándar de *sort*; éste las procesa y envía el resultado a su salida estándar, la pantalla.

Filtros importantes

En la siguiente tabla se muestran algunos de los filtros más importantes disponibles en un sistema UNIX, junto con una pequeña descripción.

Filtro	Descripción
head	Imprime las primeras líneas de un archivo
tail	Imprime las últimas líneas de un archivo
wc	Dice el número de caracteres, palabras y líneas de un archivo
sort	Clasifica alfabéticamente las líneas de un archivo
uniq	Imprime las líneas no repetidas de un archivo ya clasificado
grep	Imprime las líneas de un archivo en las que aparece una expresión
tr	Cambia o elimina caracteres de un archivo
cut	Elimina partes de cada línea de un archivo

Ejemplos

- Saber cuántos ficheros hay en `/etc`: **`ls /etc | wc -l`**
- Convertir a minúsculas todas las letras de `Fichero`: **`tr '[A-Z]' '[a-z]' <fichero`**
- Obtener las líneas 11 a 15 de `archivo`: **`head -n 15 archivo | tail -n 5`**
- Saber qué ficheros de `/etc` contienen en el nombre "conf": **`ls /etc | grep conf`**