

S en la shell Linux

Sumario

- 1 Introducción á redirección E/S
- 2 Saída estándar
- 3 Entrada estándar
- 4 Tuberías (*Pipes*)
- 5 Filtros

Introdución á redirección E/S

Neste punto, veremos os comandos que se encargan da redirección da entrada/saída.

Saída estándar

A maioría dos comandos da shell mostran os seus resultados na saída estándar, é dicir, a pantalla. Para redireccionar a saída estándar a un arquivo, empregaremos o carácter ">". Aquí vemos un exemplo:

```
[usuario@ubuntuserver]$ ls > lista_arquivos.txt
```

Neste exemplo, o comando `ls` é executado e o seu resultado é escrito no arquivo chamado `lista_arquivos.txt`. Como a saída de `ls` foi direccionada cada o arquivo, non aparece ningún resultado na pantalla.

Cada vez que o comando anterior se repite, o arquivo `lista_arquivos.txt` é sobreescrito. Se queremos que os resultados se engadan a un ficheiro xa existente temos que empregar ">>", un exemplo sería este:

```
[usuario@ubuntuserver]$ ls >> lista_arquivos.txt
```

Cando os resultados son adxuntados, os novos resultados son engadidos no final do arquivo, o arquivo faise mais largo a medida de que o o comando se repite. Se o arquivo non existise, creárase neste intre.

Entrada estándar

A entrada estándar, por defecto, é o teclado. Pero podemos redireccionar o contido dun arquivo empregando o carácter "<" do seguinte xeito:

```
[usuario@ubuntuserver]$ sort < lista_arquivos.txt
```

No exemplo anterior, empregamos o comando `sort` para procesar o contido do arquivo `lista_arquivos.txt`. Os resultados mostráranse na saída estándar, é dicir, na pantalla. Poderíamos redireccionar á saída a un arquivo do seguinte xeito:

```
[usuario@ubuntuserver]$ sort < lista_arquivos.txt > lista_arquivos_ordenados.txt
```

Por último, decir que, la única condición para utilizar las redirecciones es que se coloquen despois de todos os argumentos dos comandos empregados.

Tuberías (*Pipes*)

Normalmente, o mais empregado e potente para redireccionar a E/S es conectar varios comandos xuntos cunha tubería (*pipe*). Con tuberías, a saída estándar dun comando é a entrada do seguinte. Aquí podemos ver un exemplo:

```
[usuario@ubuntuserver]$ ls -l | less
```

Neste exemplo, a saída do comando `ls` será a entrada do comando `less`. Así, tal e como sabemos, o resultado do comando `ls -l` vaise mostrando pantalla a pantalla.

Conectando varios comandos xuntos pódense conseguir combinacións moi útiles. Aquí podemos ver varios exemplos:

```
# Mostrar os 10 primeiros arquivos do directorio actual:
```

```
[usuario@ubuntuserver]$ ls -lt | head
```

```
# Mostrar unha lista dos directorios e o espazo en disco que ocupan, ordenar de mais grande a mais pequeno:  
[usuario@ubuntuserver]$ du | sort -nr
```

```
# Mostrar o número total de arquivos no directorio actual e todos os seus directorios:  
[usuario@ubuntuserver]$ find . -type f -print | wc -l
```

Filtros

Un tipo de programas moi empregado coas tuberías son os chamados filtros. Aquí vemos varios programas que actúan como filtros e para que se empregan:

Common filter commands

Programa	Resumo do que fai
<i>sort</i>	Ordena o que recibe pola entrada estándar e mostra os resultados na saída estándar.
<i>uniq</i>	Ordena o que recibe pola entrada estándar ademais de eliminar as liñas duplicadas, mostrando logo o resultado pola saída estándar.
<i>grep</i>	Examina os datos de cada unha das liñas que recibe pola entrada estándar devolvendo todas as liñas que conteñen un conxunto de caracteres prefixados.
<i>fmt</i>	Lé texto da entrada estándar, entón mostra pola saída estándar o texto formateado.
<i>pr</i>	Toma o texto da entrada estándar e divide os datos en páxinas, cabeceiras e pes preparados para a impresión.
<i>head</i>	Devolve as primeiras liñas dun documento.
<i>tail</i>	Devolve as últimas liñas dun documento.
<i>tr</i>	Cambia caracteres. Pode ser empregado para pasar de maiúsculas a minúsculas ou ao revés.
<i>sed</i>	Trátase dun tr máis sofisticado.
<i>awk</i>	AWK é unha "linguaxe de programación" deseñado para construír potentes filtros.