

Trabajando con Ficheros

Los comandos que vamos a ver normalmente modifican los contenidos enviando el resultado a stdout, no modificando realmente el fichero original, aunque podemos redirigir o canalizar esta salida a un fichero o comando.

Comandos para combinar ficheros

Vamos a ver algunos comandos para combinar ficheros

cat

Concatena, o une el contenido, ficheros y envía el resultado a la stdout la cual podemos redireccionar a un fichero

- Concatenación de dos ficheros y redireccionar el resultado a un fichero, si el fichero concatenado existiera previamente su contenido se perdería
 - `cat fichero1 fichero2 > ficheroConcatenado`
- Concatenación de dos ficheros y redireccionar el resultado a un fichero, si el fichero concatenado existiera previamente su contenido NO se perdería
 - `cat fichero1 fichero2 >> ficheroConcatenado`
- Ver el fichero resultante, `cat` concatenará `ficheroConcatenado` con nada y el resultado lo dirige a stdout
 - `cat ficheroConcatenado`
- Crear contenido en un fichero, si el fichero existía su contenido se perderá. Tras ejecutar el siguiente comando, escribir el texto deseado, podremos utilizar `return` para crear saltos de línea, para terminar de introducir contenido pulsar `ctrl+d`
 - `cat > fichero`
- Añadir contenido a un fichero, si el fichero existía, el nuevo contenido se añadirá al final del contenido existente. Ejecutar el siguiente comando y seguir las directrices del comando anterior
 - `cat >> fichero`

`cat` cuenta con algunas opciones y modificadores, los cuales alteran los resultados producidos, como añadir números de línea, mostrar caracteres especiales,... como siempre utilizar `man` o `info` para ver la lista completa y su sintaxis

tac

Todas las explicaciones para `cat` son válidas para `tac` pero con el orden de las líneas invertido en los resultados.

join

Une dos ficheros basándose en un campo, por defecto el primero, enviando el resultado a stdout, los dos

ficheros deben estar ordenados correctamente para poder realizar una unión completa, de no ser así join realizará la unión hasta que encuentre la primera línea desordenada.

Los campos vienen delimitados por un carácter predeterminado, por defecto un espacio en blanco " ", y cada línea representa un registro, con lo que podemos tener una pequeña base de datos, con una funcionalidad limitada, pero que como práctica es interesante.

En el directorio creado anteriormente para nuestros ejemplos, creamos dos ficheros con dos campos cada uno y los siguientes contenidos, para cada uno de ellos, lo podemos hacer con cat y así practicar, o utilizar un editor GUI de nuestra distribución

fichero: uno

```
1 uno1
2 dos1
3 tres1
4 cuatro1
```

fichero: dos

```
1 uno2
1 uno21
2 dos2
3 tres2
4 cuatro2
```

unimos los dos ficheros y lo vemos en consola, fijémonos en el resultado y como maneja las entradas repetidas del segundo fichero

```
join uno dos
```

si queremos guardar el resultado utilizaremos una redirección, por ejemplo creando un nuevo fichero o reemplazando su contenido completamente

```
join uno dos > ficheroJoin
```

paste

Une dos ficheros línea a línea separándolas con una tabulación, experimenta por tu cuenta.

Comandos para la transformación de ficheros

expand

Cambia tabulaciones por espacios en blanco, puede ir muy bien para por ejemplo, modificar ficheros de código de programación, en el que se han utilizado tabulaciones para indentar el código, en estos casos con pocos anidamientos, al imprimir el código queda muy mal y se dificulta su lectura, por lo que podemos cambiar las tabulaciones por dos espacios en blanco, mejorando así su lectura/estudio.

unexpand

Lo contrario que `expand`

od

Permite ver ficheros en formato octal, aunque aplicando algunas opciones podemos verlos en formato hexadecimal, y otros atendiendo a la longitud de palabra,...

sort

Ordena el contenido del fichero/s, normalmente utiliza el primer campo y los valores ASCII, aunque tiene opciones que permiten modificar este comportamiento.

split

Permite dividir un fichero en base a unas condiciones establecidas mediante sus opciones, DEBEMOS pasarle el prefijo para los ficheros resultantes, a dicho prefijo le añadirá un conteo alfabético a cada fichero resultante, (p.e. `ficheroa`, `fichero b`, `fichero c`,...). Dependiendo de las opciones elegidas podemos dividir el fichero por bytes, líneas, ...

tr

Cambia uno a uno los caracteres que se encuentran en el grupo1 por los del grupo2 en la stdin y recordemos que podemos utilizar la redirección para realizarlo sobre un fichero, y mostrar el resultado en pantalla o a un fichero.

uniq

Elimina líneas duplicadas, útil después de ordenar un fichero con `sort` y no se quiera tener entradas duplicadas.

fmt

Principalmente formatea el ancho de las líneas de la stdin o de un fichero.

nl

Herramienta para numerar líneas, por ejemplo de un fichero de código. Tiene varias opciones interesantes.

pr

Prepara un texto para su impresión, ancho de línea, líneas por página,

Comandos para la visualización de ficheros

head

Muestra el inicio de un fichero, por defecto las 10 primeras líneas, tiene algunas opciones.

tail

Muestra el final de un fichero, por defecto las 10 ultimas líneas, tiene algunas opciones. Es especialmente útil para ver las últimas entradas de los ficheros de log, tiene opciones muy interesantes, que nos permite hacer el seguimiento de un fichero, o que lo haga hasta que termine un proceso determinado.

more

Visualizar de forma paginada un fichero

less

Visualizar de forma paginada un fichero, versión mejorada de `more` que nos permite ir tanto hacia adelante como hacia atrás, buscar en ambos sentidos, ir a una línea específica, ...

Comandos para resumir ficheros

cut

Corta las líneas entrada según lo especificado en sus opciones, y lo muestra por la salida (recordad que podemos utilizar redirecciones y utilizar ficheros)

```
cut -b2-10
```

escribir y terminar con `ctrl+d`

de cada línea devolverá desde el carácter 2 al 10, empieza a contar desde 1

wc

Cuenta las palabras, líneas, bytes,... de un fichero.

Expresiones regulares

Las expresiones regulares son cadenas que se utilizan como patrones para comparaciones

- La forma mas simple es un texto, p.e. `txt` que coincidirá con todas las cadenas que sean exactamente `txt` o aquellas que la contengan.
- Podemos hacer que coincidan con uno varios caracteres, colocándolos entre corchetes(`[]`) `[aeiou]`
- Coincide en un rango `[a-z]`
- Un carácter . p.e. `r.cord` coincidirá con cualquier cadena de 3 caracteres que empiece por "r" le siga cualquier carácter y acabe con "cord" (record, racord)
- Para indicar que la cadena debe iniciarse como indica la expresión regular, utilizaremos el `^`
- Para indicar que la cadena debe terminar como indica la expresión regular, utilizaremos el `$`
- Repetición (podemos combinarlas una combinación muy útil es . `*` que representa cualquier subcadena)
 - `?` indica cero o una aparición de la expresión regular
 - `+` indica una o mas apariciones
 - `*` cero o mas apariciones
- Una cadena u otra `|` se puede utilizar para un reemplazo como `silla|taburete` para cambiarlo

por asiento

- Para especificar como se aplican los modificadores/operadores utilizamos los () p.e. `^(silla|taburete)s`
- Si queremos utilizar alguno de los operadores propios de las expresiones regulares dentro de la cadena precederemos dicho carácter con \ p.e. `^\(((silla|taburete)s`

Existen grandes tratados sobre expresiones regulares, así que no me extendiendo mas, al menos de momento, seguro que las iremos utilizando en ejemplos.

grep

Potente herramienta para buscar coincidencias en ficheros, nos permite utilizar expresiones regulares, canalizaciones, redirecciones, recomendable dedicarle un tiempo a practicar con el y adquirir soltura.

sed

Herramienta para modificar la entrada y muestra el resultado en la stdout, se pueden utilizar redirecciones como es normal, es una herramienta compleja ver todas las opciones que nos proporciona con `man` o/y `info`

utilizar `man` o `info` para ver todas las opciones de los distintos comandos vistos

Francisco Javier López Torrijos
Analista Sistemas Informáticos de Gestión
Diseño y Desarrollo Web