

# Comandos Habituales con Ficheros de Texto

cat • tac • tr • cut • wc • head • tail

Tuberías | “ Comillas " " ' '

```
user@linux:~$ text_processing _
```

# | Tuberías | (Pipe)

## ¿Qué es?

Conecta la **salida estándar** de un comando con la **entrada estándar** de otro

## <> Sintaxis

```
comando1 | comando2
```

## ★ Características

- Encadena comandos en una sola línea
- Permite procesar datos en etapas
- Fundamental en scripts de shell

## 📄 Ejemplos

Filtrar contenido de archivo

```
cat archivo.txt | grep patron
```

Muestra líneas que coinciden con "patron"

Contar archivos y directorios

```
ls | wc -l
```

Cuenta elementos del directorio actual

Extraer líneas específicas

```
head -10 archivo.txt | tail -5
```

Muestra líneas 6 a 10 del archivo

# | Comillas: "" y ''

## ” Comillas Dobles ""

Permiten **expansión de variables** y **caracteres especiales**

### \$ Expansión de Variables

\$VARIABLE se reemplaza por su valor

### ` Sustitución de Comandos

`comando` ejecuta y reemplaza con la salida

### \ Caracteres de Escape

\n, \t, \\$ son interpretados

Ejemplo

```
echo "$HOME"
```

```
/home/usuario
```

## ” Comillas Simples ''

**Todo es literal**, sin interpretación de caracteres especiales

### ⊘ Sin Expansión

\$VARIABLE se muestra literalmente

### ⊘ Sin Ejecución

`comando` no se ejecuta

### ⊘ Sin Escape

\n, \t, \\$ se muestran literalmente

Ejemplo

```
echo '$HOME'
```

```
$HOME
```

# | Comando cat

## Descripción

Concatena y muestra el contenido de archivos

## <> Sintaxis

```
cat [opciones] archivo
```

## Opciones

- n** Numera todas las líneas
- b** Numera solo líneas no vacías
- s** Comprime líneas vacías consecutivas
- E** Muestra \$ al final de cada línea
- v** Muestra caracteres no imprimibles
- T** Muestra tabulaciones como ^I

## Ejemplos

Mostrar contenido de archivo

```
cat archivo.txt
```

Muestra el contenido completo del archivo

Con numeración de líneas

```
cat -n documento.txt
```

Muestra contenido con números de línea

Concatenar archivos

```
cat archivo1.txt archivo2.txt > combinado.txt
```

Combina dos archivos en uno nuevo

# | Comando tac

## Descripción

Muestra el contenido de archivos en **orden inverso** (última línea primero)

## <> Sintaxis

```
tac [archivo]
```

## Características

- Invierte el orden de las líneas
- No cambia el orden de palabras o letras
- Útil para ver logs recientes primero
- Compatible con redirección (>, >>)

## Ejemplos

Mostrar archivo invertido

```
tac archivo.txt
```

Muestra líneas de abajo a arriba

Invertir múltiples archivos

```
tac archivo1.txt archivo2.txt
```

Invierte cada archivo por separado

Combinar con pipe

```
cat archivo.txt | tac
```

Muestra contenido en orden inverso

# | Comando tr



## Descripción

Traduce, elimina o comprime caracteres de la entrada estándar

## <> Sintaxis

```
tr [opciones] conjunto1  
[conjunto2]
```



## Opciones

- d Elimina caracteres en conjunto1
- s Comprime caracteres repetidos
- t Trunca conjunto1 a longitud de conjunto2
- c Complementa conjunto1 (selecciona lo contrario)



## Ejemplos

Convertir a mayúsculas

```
echo 'hola' | tr 'a-z' 'A-Z'
```

Resultado: HOLA

Eliminar vocales

```
tr -d 'aeiou' < archivo.txt
```

Elimina todas las vocales del archivo

Comprimir espacios

```
tr -s ' ' < archivo.txt
```

Convierte múltiples espacios en uno solo

# | Comando cut

## Descripción

Corta secciones de cada línea de archivos (columnas o caracteres)

## <> Sintaxis

```
cut [opciones] archivo
```

## Opciones

- d** Especifica delimitador
  - f** Selecciona campos
  - c** Selecciona caracteres
  - b** Selecciona bytes
  - Cambia delimitador de salida
- output-delimiter**

## Ejemplos

Extraer campos específicos (CSV)

```
cut -d':' -f1,3 archivo.csv
```

Muestra campos 1 y 3 separados por :

Extraer caracteres específicos

```
cut -c1-10 archivo.txt
```

Muestra caracteres del 1 al 10

Extraer segundo campo (espacio)

```
cut -d' ' -f2 archivo.txt
```

Muestra segunda columna separada por espacio

# | Comando wc



## Descripción

Cuenta líneas, palabras y caracteres de archivos



## Sintaxis

```
wc [opciones] archivo
```



## Opciones

- l** Cuenta líneas
- w** Cuenta palabras
- m** Cuenta caracteres
- c** Cuenta bytes
- L** Longitud de línea más larga



## Ejemplos

Contar todo (líneas, palabras, caracteres)

```
wc archivo.txt
```

Salida: 15 127 789 archivo.txt

Contar solo líneas

```
wc -l archivo.txt
```

Salida: 15 archivo.txt

Contar archivos en directorio

```
ls | wc -l
```

Número de archivos y directorios



# | Comando head



## Descripción

Muestra las primeras líneas de un archivo  
(por defecto 10 líneas)



## Sintaxis

```
head [opciones] archivo
```



## Opciones

- n N** Muestra N primeras líneas
- c N** Muestra N primeros bytes
- q** No muestra nombre de archivo
- v** Siempre muestra nombre de archivo



## Ejemplos

Mostrar primeras 10 líneas

```
head archivo.txt
```

Visualiza inicio del archivo

Mostrar primeras 5 líneas

```
head -n 5 archivo.txt
```

Solo las primeras 5 líneas

Extraer líneas 11-20

```
head -n 20 archivo.txt | tail -n 10
```

Líneas 11 a 20 del archivo

# | Comando tail

## Descripción

Muestra las últimas líneas de un archivo (por defecto 10 líneas)

## <> Sintaxis

```
tail [opciones] archivo
```

## Opciones

- n N** Muestra N últimas líneas
- c N** Muestra N últimos bytes
- f** Sigue archivo en tiempo real
- F** Sigue incluso si archivo es rotado
- q** No muestra nombre de archivo
- v** Siempre muestra nombre de archivo

## Ejemplos

Mostrar últimas 10 líneas

```
tail archivo.txt
```

Visualiza final del archivo

Seguir log en tiempo real

```
tail -f /var/log/syslog
```

Muestra nuevas líneas a medida que se escriben

Mostrar desde línea 50 en adelante

```
tail -n +50 archivo.txt
```

Líneas 50 hasta el final del archivo