

U4. Sistema de fitxers. Ubuntu (III)

TRACTAMENT DE TEXT (I)

@tofermos 2024

Índice

Resum	2
1. grep	2
Modificadors essencials:	2
Exemple:	2
2. tr	3
Modificadors essencials:	3
Exemple:	3
3. cut	3
Modificadors essencials:	3
Exemple:	4
4. sort	4
Modificadors essencials:	4
Exemple:	4
5 wc	5
Exemples d'ús de wc	5
Ús de wc amb combinacions de pipes ()	6

Resum

Tot seguit coneixerem unes ordres de Linux que ens serveixen per al tractament de fitxers de text pla. En el món de Linux són molt útils a l'hora de consultar o fer modificacions en fitxers de configuració del sistema i també per consultar altres fitxers de registres sobre estats o l'èxida a pantalla que donen les ordres de Linux.

1. grep

L'ordre **grep** s'utilitza per buscar text dins de fitxers. Mostra línies que coincideixen amb un patró específic.

Modificadors essencials:

- **-i**: ignora majúscules i minúscules.
- **-v**: mostra línies que **no** coincideixen amb el patró.
- **-r**: busca de manera recursiva dins de directoris.
- **-n**: mostra el número de línia on es troba la coincidència.

Exemple:

```
# Busca "error" en el fitxer log.txt, ignorant majúscules i minúscules  
grep -i "error" log.txt
```

```
# Busca "warning" recursivament dins el directori logs i mostra el número de línia  
grep -r -n "warning" logs/
```

```
# Busca línies que comencen amb "Start" en el fitxer text.txt  
grep "^Start" text.txt
```

```
# Busca línies que acaben amb "End" en el fitxer text.txt  
grep "End$" text.txt
```

2. tr

L'ordre `tr` s'utilitza per traduir o suprimir caràcters. Generalment es fa servir per convertir majúscules a minúscules, eliminar espais, etc.

Modificadors essencials:

- `-d`: suprimeix els caràcters especificats.
- `-s`: substitueix seqüències repetides del mateix caràcter amb una sola instància.

Exemple:

```
# Converteix majúscules a minúscules
echo "HELLO WORLD" | tr 'A-Z' 'a-z'

# Suprimeix tots els espais
echo "Hello  World" | tr -d ' '

# Substitueix seqüències repetides d'espais amb un sol espai
echo "Hello    World" | tr -s ' '

```

3. cut

L'ordre `cut` s'utilitza per seleccionar parts de cada línia d'un fitxer o flux d'entrada (com ara una columna específica d'un CSV).

Modificadors essencials:

- `-d`: defineix el delimitador (per defecte és el tabulador).
- `-f`: selecciona els camps específics.

Exemple:

```
# Mostra la primera columna (assumint que és un CSV)  
cat dades.csv | cut -d ',' -f 1  
  
# Mostra la segona i tercera columna separades per espais  
echo "nom edat ciutat" | cut -d ' ' -f 2,3
```

4. sort

L'ordre `sort` s'utilitza per ordenar les línies d'un fitxer o entrada. Es pot utilitzar amb diversos criteris d'ordenació.

Modificadors essencials:

- `-r`: ordena en ordre invers (descendent).
- `-n`: ordena numèricament en lloc d'alfabèticament.
- `-k`: especifica una columna per ordenar.

Exemple:

```
# Ordena el fitxer numèricament (de menor a major)  
sort -n nombres.txt  
  
# Ordena el fitxer en ordre descendent  
sort -r paraules.txt  
  
# Ordena el fitxer per la segona columna  
sort -k 2 dades.txt
```

5 wc

L'ordre `wc` (“word count” o comptador de paraules) serveix per **comptar línies, paraules i caràcters (bytes)** en un fitxer o a l'eixida d'una altra ordre. Té diversos modificadors que permeten filtrar quina informació es vol veure. Aquestes opcions són:

- `-l`: Comptar el nombre de línies.
- `-w`: Comptar el nombre de paraules.
- `-c`: Comptar el nombre de caràcters (bytes).
- `-m`: Comptar el nombre de caràcters (tenint en compte caràcters multibyte com accents i símbols).
- `-L`: Mostrar la línia més llarga en nombre de caràcters.

Exemples d'ús de `wc`

1. Comptar línies en un fitxer

```
wc -l fitxer.txt
```

Això mostrarà el nombre de línies en el fitxer `fitxer.txt`.

2. Comptar paraules en un fitxer

```
wc -w fitxer.txt
```

Això mostrarà el nombre de paraules en el fitxer `fitxer.txt`.

3. Comptar caràcters en un fitxer

```
wc -c fitxer.txt
```

Això mostrarà el nombre de caràcters (bytes) en el fitxer `fitxer.txt`.

4. Comptar caràcters multibyte en un fitxer

```
wc -m fitxer.txt
```

Aquest comptarà tots els caràcters, tenint en compte els multibyte com accents o símbols especials.

5. Mostrar la línia més llarga en nombre de caràcters

```
wc -L fitxer.txt
```

Aquest comptarà els caràcters de la línia més llarga del fitxer `fitxer.txt`.

6. Mostrar totes les estadístiques a la vegada

```
wc fitxer.txt
```

Això mostrarà el nombre de línies, paraules i bytes del fitxer `fitxer.txt` en un sol ordre.

Ús de `wc` amb combinacions de pipes (|)

L'ús de `wc` combinat amb altres ordres `cat` o `grep` és prou habitual per filtrar o comptar el nombre d'ocurrències d'un patró.

1. **Comptar el nombre de línies que contenen una paraula específica amb `grep` i `wc`**

```
grep "paraula" fitxer.txt | wc -l
```

Aquesta ordre cerca la paraula “paraula” en el fitxer `fitxer.txt` i després compta el nombre de línies que contenen aquesta paraula.

2. **Comptar el nombre de paraules en diversos fitxers amb `cat` i `wc`**

```
cat fitxer1.txt fitxer2.txt | wc -w
```

Aquest combinat mostra el nombre total de paraules en els fitxers `fitxer1.txt` i `fitxer2.txt` sumades.

3. **Comptar línies d'un fitxer excloent línies buides amb `grep` i `wc`**

```
grep -v "^$" fitxer.txt | wc -l
```

Aquesta ordre cerca totes les línies no buides (^\$ significa línies buides) en `fitxer.txt` i compta el nombre de línies no buides trobades.

Aquests exemples mostren com utilitzar `wc` de diferents maneres per analitzar fitxers i també com combinar-lo amb altres ordres mitjançant pipelines per obtenir informació més específica i detallada.

4. **opció `-c` del `grep`**

Com a curiositat podem veure l'opció -c del grep.

```
tomas@portatil:~$ grep ^tomas /etc/passwd
tomas:x:1000:1000:tomas,,,:/home/tomas:/bin/bash
tomas@portatil:~$ grep ^tomas /etc/passwd -c
1
tomas@portatil:~$ grep ^tomas /etc/passwd |wc -l
1
```