

GABRIELE ROMANATO

Uso del comando stat in Bash

Il comando `stat` è uno strumento molto utile in Bash, utilizzato per visualizzare informazioni dettagliate su file e directory. Fornisce dati come dimensione, permessi, timestamp, e molto altro, che spesso non sono immediatamente accessibili con comandi di base come `ls`. In questo articolo esploreremo come utilizzare `stat`, con esempi pratici per aiutarti a sfruttare al meglio le sue funzionalità.

La sintassi di base del comando `stat` è:

```
stat [opzioni] [file/directory]
```

Dove:

- `[opzioni]` rappresenta eventuali parametri che si possono passare per personalizzare l'output.
- `[file/directory]` è il file o la directory di cui vuoi ottenere informazioni.

Output predefinito:

```
File: test.txt
Size: 1234          Blocks: 8          IO Block: 4096
regular file
Device: 802h/2050d  Inode: 395978      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/  user)   Gid: ( 1000/  user)
Access: 2024-10-05 09:35:12.000000000 +0200
Modify: 2024-10-05 09:32:45.000000000 +0200
```

```
Change: 2024-10-05 09:32:45.000000000 +0200
Birth: -
```

Spiegazione dell'output:

- **File:** nome del file.
- **Size:** dimensione del file in byte.
- **Blocks:** numero di blocchi allocati sul disco per questo file.
- **IO Block:** dimensione del blocco di input/output del filesystem.
- **Device:** ID del dispositivo su cui risiede il file.
- **Inode:** numero dell'inode del file.
- **Links:** numero di hard link associati al file.
- **Access:** permessi del file in formato ottale e simbolico, insieme a UID e GID del proprietario.
- **Access/Modify/Change:** timestamp di accesso, modifica, e cambiamento delle proprietà del file.
- **Birth:** la data di creazione del file, se supportata dal filesystem.

Uno degli aspetti più potenti del comando `stat` è la possibilità di personalizzare l'output usando il parametro `--format o -c`. Questo permette di specificare esattamente quali informazioni vogliamo visualizzare e come.

Ecco un esempio in cui visualizziamo solo la dimensione e i permessi del file:

```
stat --format="Dimensione: %s bytes, Permessi: %A" test.txt
```

L'output sarà simile a:

```
Dimensione: 1234 bytes, Permessi: -rw-r--r--
```

Qui, %s viene utilizzato per ottenere la dimensione del file e %A per i permessi in formato leggibile.

Ecco un elenco di specificatori di formattazione comunemente utilizzati con il comando `stat`:

- %s: Dimensione del file in byte.
- %A: Permessi del file in formato leggibile.
- %a: Permessi del file in formato ottale.
- %U: Nome dell'utente proprietario del file.
- %G: Nome del gruppo proprietario.
- %F: Tipo di file (es. "directory", "regular file").
- %Y: Timestamp di ultima modifica, in formato Unix.
- %x: Data e ora dell'ultimo accesso.
- %y: Data e ora dell'ultima modifica.
- %z: Data e ora dell'ultimo cambiamento delle proprietà.

Ad esempio, per visualizzare la dimensione, i permessi e l'ultimo tempo di modifica di un file:

```
stat --format="Dimensione: %s, Permessi: %A, Modificato: %y"  
test.txt
```

Se vuoi ottenere solo i permessi di un file, puoi utilizzare il comando `stat` con la formattazione personalizzata:

```
stat -c "%A" test.txt
```

Oppure, per ottenere i permessi in formato ottale:

```
stat -c "%a" test.txt
```

Il comando `ls -l` può sembrare simile a `stat`, ma ci sono differenze significative. Mentre `ls` fornisce una panoramica più generale e più adatta alla visualizzazione rapida di file, `stat` fornisce informazioni molto più dettagliate su singoli file o directory. Inoltre, `ls` non mostra alcune informazioni come i numeri di inode o il numero di blocchi allocati.

Conclusione

Il comando `stat` è uno strumento estremamente potente per chiunque lavori con file e directory in un ambiente Bash. Che tu voglia sapere i permessi, i timestamp o i dettagli sul filesystem, `stat` ti fornisce tutte le informazioni necessarie in modo efficiente e personalizzabile.