Uso batch dell'editor `vi`

mobytrick

8 settembre 2014

È possibile usare uno strumento eminentemente interattivo come l'editor in modalità batch, ad esempio in uno script? La risposta è affermativa. A questo proposito appunto qui alcune chicche che potrebbero essere interessanti.

L'editor principe nel mondo Unix/Linux è il `vi`, **vi**sual editor. O, meglio, era perché soppiantato da `vim`, la versione *improved*. C'è una leggera differenza all'invocazione dell'editor. `vim` esegue i comandi contenuti nel file `~/.viminfo.rc`, se esiste. Se non c'è ripiega su `~/.exrc`. Se nemmeno questo esiste, opera con le impostazioni di default. vi eseguiva i comandi del file `~/.exrc`, se esisteva. In mancanza di questo, operava con le impostazioni di default.
Una delle innumerevoli configurazioni possibili riguarda la numerazione delle righe, normalmente non attivata. Nell'ipotesi che manchi qualsiasi file di configurazione, si possono seguire due vie: una *interna* che agisce sul comportamento di `vi` emntre questo è attivo, l'altra *preventiva*, perché agisce sulle configurazioni di `vi`. Con la prima modalità, una volta aperto l'editor, per modificare le impostazioni si agisce dalla modalità *Command Line*. Per passare a questa è sufficiente pigiare il tasto `:` (doppio punto). Il cursore si sposta in fondo della schermata, a sinistra ove è possibile l'immissione dei comandi. Quello da usare è `set number` (abbreviabile in `set nu`) per attivare la numerazione e `set nonumber` (abbreviabile in `set nonu`) per disattivarla. Agendo, preventivamente, la numerazione può essere attivata—disattivata in fase di richiamo dell'editor scegliendo tra 3 annotazioni quasi equivalenti:

1. `vi -c "set nu"` file
2. `vi "+set nu"` file
3. `vi -cmd "set nu"` file

Usando le prime due sintassi il comando viene eseguito **dopo** l'apertura del file. La terza forma li esegue **prima**. La documentazione riporta che si possono passare in linea sino a 10 comandi.

Esiste però il modo di far eseguire veri e propri comandi. Supponiamo di avere un file costituito da stringhe di caratteri, una per riga, ed ordinate in modo casuale. Volendo ordinare alfabeticamente il contenuto del file si procede nel seguente modo. Ci si accerta di non essere in modalità `Insert` (basta premere il tasto `Esc[ape]`). Quindi si digitano i seguenti caratteri:

```
:%!sort
```

che si interpretano così:

verb;:;	passaggio in Command Mode
`%`	seleziona tutto il file
`!`	a quanto selezionato applicare un comando Unix
`sort`	comando da applicare

Infine si esce in modo da rendere le modifiche effettive. Il comando è:

```
:wq
```

I caratteri si interpretano così:

`:`	passaggio in Command Line
`w`	write
`q`	quit

Ora è possibile operare in questo modo. I 2 comandi vengono scritti in un file di appoggio, sia questo `/tmp/comandi`, e poi si procede così:

```
vi -S /tmp/comandi file
```

I comandi vengono eseguiti e siccome questi comprendono anche l'uscita non si ha la percezione del lavoro che viene svolto. Non resta che ispezionare il contenuto del file per constatare che ora il contenuto è ordinato. È chiaro che il file di appoggio può essere costruito ad hoc con la tecnica degli *here documents* oppure metodi equivalenti. In tal modo è possibile usare l'editor in uno script. A mo' d'esempio si riporta lo script completo che crea sia il file da ordinare che quello dei comandi ed il loro uso.

```
#!/bin/bash
rm -f file
cat >> file << EOF
uno
due
tre
Tizio
Caio
Sempronio
EOF
rm -f comandi
cat >> comandi << EOF
:%! sort
ZZ
EOF
echo -e "\tfile originale"
cat file
vi -S comandi file >
/dev/null 2> /dev/null
echo -e "\tfile ordinato"
cat file
rm -f comandi file
exit 0
```

Sommariamente. Con la tecnica degli *here documents* viene creato -e successivamente visualizzato- un file disordinato. Poi viene creato il file dei comandi. Segue la sua applicazione al file. Quindi viene rimostrato il contenuto del file, in modo da accertarsi che l'ordinamento sia stato effettivamente effettuato. Questo è il risultato facendo eseguire lo script.

```
        file originale
uno
due
tre
Tizio
Caio
Sempronio

        file ordinato
Caio
due
```

```
Sempronio
Tizio
tre
uno
```

Talvolta alcune delle operazioni effettuate in una sessione di editing vengono eseguite quasi a memoria e potrebbe essere difficoltoso scriverle. C'è la possibilità di trarsi d'impaccio facendole registrare in un file dall'editor stesso. Ricapitolando:

`cp file succedaneo`	creazione copia
`vi -w log_comandi succedaneo`	registrazione dei comandi
`vi -S log_comandi file`	applicazione dei comandi
`rm -f succedaneo`	rimozione copia

Per prima cosa si fa una copia temporanea del file. E si lavora su questa. Tutti i comandi impartiti, compresi gli errori, vengono scritti nel file indicato da `-w`. File che poi potrà essere applicato, nel modo descritto prima, al file vero e proprio. Attenzione alla sua struttura perché non contiene il carattere `\n` (new line) bensì `\r` (carriage return).