

\*usr\_32.txt\* Per Vim version 8.2. Ultima modifica: 2010 Jul 20

VIM USER MANUAL - di Bram Moolenaar

L'albero degli undo  
Traduzione di questo capitolo: Antonio Colombo

Vim gestisce un undo multiplo. Se tornate indietro con alcune modifiche e poi fate delle nuove modifiche, create un ramo nell'albero [nella storia] degli undo. Questo capitolo riguarda i movimenti fra i rami di quest'albero.

<b>32.1</b> <b>32.2</b> <b>32.3</b> <b>32.4</b>	Undo fino a una scrittura file Numerazione modifiche Percorrere la storia delle modifiche Viaggiare nel tempo
--	--

Capitolo seguente: Capitolo precedente: Indice:	<b>usr_40.txt</b> <b>usr_31.txt</b> <b>usr_toc.txt</b>	Definire nuovi comandi Sfruttare la GUI
---	--	--

=====

\*32.1\* Undo fino a una scrittura file

Talora fate numerose modifiche, e poi vi accorgete di voler tornare al punto in cui avete salvato il file per l'ultima volta. Potete farlo con questo comando: >

```
:earlier lf
```

La "f" significa qui "file".

Si può ripetere questo comando per andare ulteriormente indietro nel tempo. Oppure usare un contatore diverso da 1 per tornare indietro più velocemente.

Se siete andati indietro troppo, tornate in avanti con: >

```
:later lf
```

Nota: questi comandi funzionano in base al tempo trascorso. Questo è importante se avete fatto delle modifiche dopo essere tornati indietro nel tempo (con altre modifiche). La cosa è spiegata nella prossima sezione.

Notate anche che stiamo parlando di scritture del testo. Per le scritture di informazioni di undo in un file, vedere **undo-persistence**.

=====

\*32.2\* Numerazione modifiche

Nella sezione **02.5** abbiamo trattato solo una linea di undo/redo. Ma è anche possibile percorrere altre linee. Questo capita quando si annullano alcune modifiche (undo) e poi si fanno modifiche differenti (dal redo). Queste nuove modifiche generano dei rami nell'albero degli undo.

Cominciamo col testo "uno". La prima modifica fatta è aggiungere " sue". Poi torniamo alla prima 's' e la cambiamo in 'd'. Abbiamo finora due modifiche, numerate 1 e 2, e tre versioni del testo:

```

      uno ~
      |
    modifica 1
      |
    uno sue~
      |
    modifica 2
      |
    uno due ~
  
```

facendo undo una volta si torna a "uno sue", e cambiando "uno" con "cose" si crea un ramo nell'albero degli undo:

```

      uno ~
      |
  
```

```

      modifica 1
      |
    uno sue ~
    /   \
  modifica 2   modifica 3
    |           |
  uno due     cose sue ~

```

Si può ora usare il comando `|u|` per fare undo. Usandolo due volte, si torna a "uno". Se si usa `|CTRL-R|` per rifare, si va a "uno sue". Un ulteriore `|CTRL-R|` vi porta a "cose sue". Quindi undo e redo vanno su e giù nell'albero, passando dal percorso utilizzato più recentemente.

Quel che importa qui è l'ordine in cui le modifiche sono fatte. Undo e redo non sono considerate modifiche, in questo contesto. Dopo ogni modifica si forma una nuova versione del testo.

Nota Si noti che sono le modifiche sono numerate, il testo che si vede nell'albero qui sopra non ha alcun identificativo. I testi sono identificati molto spesso dal numero della modifica sopra di essi. Talora invece sono identificati dal numero di una delle modifiche che stanno sotto di loro, specie quando ci si muove verso l'alto dell'albero, in modo da sapere quale modifica è stata appena annullata.

---

### \*32.3\* Percorrere la storia delle modifiche

Come si fa ad arrivare a "uno due" adesso? Potete usare questo comando: >

```
:undo 2
```

Il testo è ora "uno due", siete sotto la modifica 2. Si può usare il comando `|:undo|` per saltare sotto ogni modifica nell'albero.

Ora fate un'altra modifica: cambiate "uno" in "noi":

```

      uno ~
      |
    modifica 1
      |
    uno sue ~
    /   \
  modifica 2   modifica 3
    |           |
  uno due     cose sue ~
    |
  modifica 4
    |
  noi due ~

```

A questo punto cambiate idea e volete tornare a "uno sue". Usate il comando `|g-|`. Questo comando torna indietro nel tempo. Ovvero non percorre l'albero verso l'alto o verso il basso, ma percorre le modifiche andando indietro in ordine di tempo.

Se date più volte il comando `|g-|` vedrete il testo cambiare così:

```

cose sue ~
uno due ~
uno sue ~
uno ~

```

Usando `|g+|` vi muovete invece in avanti nel tempo:

```

uno ~
uno sue ~
uno due ~
cose sue ~
noi due ~

```

Usare `|:undo|` è utile se sapete a quale modifica volete saltare. `|g-|` e `|g+|` sono utili se non sapete esattamente quale è il numero di una modifica.

Potete inserire un contatore prima di `|g-|` e `|g+|` per eseguirli più volte.

---

\*32.4\* Viaggiare nel tempo

Dopo che avete lavorato ad un testo per un po' di tempo, l'albero comincia a diventare grande.

A quel punto potreste voler tornare al testo di alcuni minuti prima.

Per vedere che rami ci sono nell'albero degli undo, usate questo comando: >

```
      :undolist
<      numero modif.   ora ~
          3           2 16 secondi fa
          4           3  5 secondi fa
```

Qui potete vedere il numero di foglie in ogni ramo, e quando è stata fatta la modifica. Assumendo che siamo sotto la modifica 4, a "noi due", si può tornare indietro di dieci secondi con questo comando:

```
:earlier 10s
```

A seconda del tempo che avete impiegato a fare le modifiche, vi ritroverete in una certa posizione nell'albero. L'argomento per il comando |:earlier| può essere espresso con "m" per indicare i minuti, ed "h" per le ore. Per tornare indietro il più possibile usate un numero grande: >

```
:earlier 10h
```

Per viaggiare ancora in avanti nel tempo usate il comando |:later|: >

```
:later 1m
```

Gli argomenti sono "s", "m" e "h", ovvero gli stessi di |:earlier|.

---

Capitolo seguente: |usr\_40.txt| Definire nuovi comandi

Copyright: vedere |manual-copyright| vim:tw=78:ts=8:ft=help:norl:

Per segnalazioni scrivere a vimdoc.it at gmail dot com  
oppure ad Antonio Colombo azcl00 at gmail dot com