

\*usr\_11.txt\* Per Vim version 8.0. Ultima modifica: 2018 Apr 13

VIM USER MANUAL - di Bram Moolenaar  
Traduzione di questo capitolo: Roberto Franceschini

### Recupero dopo un blocco

Il vostro computer si è bloccato? E voi avevate trascorso delle ore a scrivere? Niente paura! Vim immagazzina informazioni sufficienti a recuperare la maggior parte del vostro lavoro. Questo capitolo mostra come recuperare il vostro lavoro e spiega come viene usato il file di swap.

|11.1| Fondamenti del recupero  
|11.2| Dove si trova il file di swap?  
|11.3| Bloccato o no?  
|11.4| Altre letture

Capitolo seguente: |usr\_12.txt| Trucchi ingegnosi  
Capitolo precedente: |usr\_10.txt| Fare grandi modifiche  
Indice: |usr\_toc.txt|

#### \*11.1\* Fondamenti del recupero

Nella maggior parte dei casi recuperare un file è semplice, se conoscete quale file stavate scrivendo (e se il disco fisso funziona ancora). Aprite quel file con Vim aggiungendo l'argomento "-r": >

```
vim -r help.txt
```

Vim leggerà il file di swap (usato per memorizzare il testo che stavate scrivendo) e potrà leggere vari pezzi del file originale. Se Vim è in grado di recuperare le vostre modifiche, vedrete questi messaggi (naturalmente i nomi di file saranno diversi):

```
Uso swap file ".help.txt.swp" ~
File originale "~/vim/runtime/doc/help.txt" ~
Recupero completato. Dovresti controllare se va tutto bene. ~
(Potresti salvare questo file con un altro nome ed eseguire ~
'diff' rispetto al file originale per vedere le differenze) ~
È consigliato cancellare il file .swp adesso. ~
```

Per sicurezza, salvate questo file con un nome diverso: >

```
:write help.txt.recuperato
```

Confrontate il file con l'originale per verificare di aver ottenuto quello che vi aspettavate. Vimdiff è molto utile per ciò |08.7|. Per esempio: >

```
:write help.txt.recuperato
:diffsp help.txt
```

Verificate che l'originale non contenga una versione più recente di quella recuperata (se avete salvato il file poco prima che il computer si bloccasse). E controllate che non manchino righe (qualcosa è andato male e Vim non è quindi riuscito a recuperare interamente).

Se Vim produce dei messaggi di avviso durante il recupero, leggeteli attentamente. Comunque ciò capita raramente.

Se il ripristino ha prodotto un testo uguale al contenuto corrente del file riceverete il seguente messaggio:

```
Uso swap file ".help.txt.swp" ~
File originale "~/vim/runtime/doc/help.txt" ~
Ripristino terminato. Il contenuto del buffer coincide con quello ~
del file. ~
È consigliato cancellare il file .swp adesso. ~
```

Ciò capita se avete già recuperato le vostre modifiche, o se avete riscritto il file dopo averlo modificato. È ora buona cosa cancellare il file di swap.

È normale che gli ultimi pochi cambiamenti non possano essere recuperati.

Vim invia i cambi al disco quando non scrivete per circa quattro secondi, o dopo che avete digitato circa duecento caratteri. Questo è impostato con le opzioni 'updatetime' e 'updatecount'. Così quando Vim non ha avuto la possibilità di salvare quando il sistema si è bloccato, i cambi successivi all'ultimo invio saranno persi.

Se stavate scrivendo senza un nome di file, date una stringa vuota come argomento: >

```
vim -r ""
```

Dovete essere nella directory giusta, altrimenti Vim non può trovare il file di swap.

=====

**\*11.2\*** Dove si trova il file di swap?

Vim può memorizzare il file di swap in molti posti. Normalmente è nella stessa directory del file originale. Per trovarlo posizionatevi nella directory del file e usate: >

```
vim -r
```

Vim elencherà i file di swap che può trovare. Guarderà anche in altre directory dove potrebbero trovarsi i file di swap per i file della directory corrente. Comunque non troverà file di swap presenti in altre directory, e non controllerà l'albero delle directory.

Il risultato potrebbe apparire come questo:

```
Swap file trovati: ~
Nella directory in uso: ~
1.   .main.c.swp ~
    proprietario: mool   datato: Tue May 29 21:00:25 2001 ~
    nome file: ~mool/vim/vim6/src/main.c ~
    modificato: YES ~
    nome utente: mool   nome computer: masaka.moolenaar.net ~
ID del processo: 12525 ~
Nella directory ~/tmp: ~
-- nessuno -- ~
Nella directory /var/tmp: ~
-- nessuno -- ~
Nella directory /tmp: ~
-- nessuno -- ~
```

Se ci sono più file che possono sembrare quello che cercate, vi verrà presentato un elenco e vi verrà richiesto di digitare il numero di quello che volete usare. Guardate con attenzione la data per decidere quale utilizzare.

Se non sapete quale usare, provateli uno alla volta e verificate se sono quello cercato.

#### UTILIZZARE UN FILE DI SWAP SPECIFICO

Se sapete quale file di swap utilizzare, lo potete recuperare dando il nome del file di swap. Vim troverà il nome del file originale leggendo il file di swap.

Esempio: >

```
vim -r .help.txt.swo
```

Questo è utile anche quando il file di swap è in una directory diversa da quella che ci si aspetta. Vim riconosce come swap file tutti i file il cui nome segua il modello \*.s[uvw][a-z].

Se anche questo non funziona, guardate quali nomi di file Vim restituisce e rinominateli di conseguenza. Controllate l'opzione 'directory' per sapere dove Vim può aver messo il file di swap.

#### Nota:

Vim prova a trovare il file di swap cercando nelle directory contenute nell'opzione 'dir' i file che verificano la stringa "filename.sw?".

Se l'espansione della 'wildcard' (metacarattere) non funziona (ad esempio quando l'opzione 'shell' non è valida), Vim fa un

tentativo disperato per trovare il file "filename.swp". Se anche questo fallisce, dovreste dare il nome completo del file di swap per riuscire a recuperare il file.

```
=====
*11.3*   Bloccato o no?                                *ATTENTION* *E325*
```

Vim prova ad impedirvi di fare stupidaggini. Supponiamo che abbiate innocentemente iniziato a modificare un file e vi aspettiate di visualizzarne il contenuto. Invece Vim produce un lungo messaggio di questo tipo:

```

E325: ATTENZIONE ~
Trovato uno swap file di nome ".main.c.swp" ~
  proprietario: mool  datato: Tue May 29 21:09:28 2001 ~
    nome file: ~mool/vim/vim6/src/main.c ~
    modificato: no ~
    nome utente: mool  nome computer: masaka.moolenaar.net ~
  ID del processo: 12559 (ancora attivo) ~
Mentre aprivo file  "main.c" ~
    datato: Tue May 29 19:46:12 2001 ~
~
(1) Un altro programma può essere in edit sullo stesso file. ~
    Se è così, attenzione a non trovarvi con due versioni ~
    differenti dello stesso file a cui vengono apportate modifiche. ~
    Esci, o continua con prudenza. ~
~
(2) Una sessione di edit per questo file è finita male. ~
    Se è così, usa ":recover" oppure "vim -r main.c" ~
    per recuperare modifiche fatte (vedi ":help recovery"). ~
    Se hai già fatto ciò, cancella il file di swap ".help.txt.swp" ~
    per non ricevere ancora questo messaggio. ~
```

Ricevete questo messaggio perché, quando iniziate a modificare un file, Vim controlla se un file di swap per esso esista già. Se ne trovasse uno deve esserci qualcosa di sbagliato. Si può verificare una di queste due situazioni:

1. Un'altra sessione di modifica è attiva per questo file. Cercate nel messaggio la linea con "process ID". Può apparire così:

```
ID del processo: 12559 (ancora attivo) ~
```

Il testo "(ancora attivo)" indica che il processo che sta utilizzando questo file gira sullo stesso computer. Se state lavorando su un sistema non-Unix non avrete questa informazione. Se state utilizzando questo file in rete non potete vedere questa informazione, in quanto il processo potrebbe girare su un altro computer. In questi due casi dovreste capire da soli in che situazione vi trovate.

Se un'altra sessione di Vim stesse editando lo stesso file, continuare nella modifica porterà ad averne due versioni. Quella che verrà salvata per ultima sovrascriverà la prima, e le modifiche di questa andranno perse. In questo caso sarebbe opportuno chiudere subito Vim.

2. Il file di swap potrebbe essere il risultato di un blocco precedente di Vim o del computer. Verificate i dati contenuti nel messaggio. Se la data del file di swap è successiva a quella del file che state modificando, ed è presente questa linea:

```
modificato: YES ~
```

Allora molto probabilmente esisterà una sessione di modifica risalente a poco prima del crash che varrebbe la pena di salvare.

Se la data del file è successiva a quella del file di swap, esso può essere stato modificato dopo il blocco (forse l'avete già recuperato, senza poi cancellare il file di swap?), oppure il file è stato salvato prima del blocco, ma dopo l'ultima scrittura sul file di swap (in questo caso siete fortunati, il vecchio file di swap non vi serve neppure). Vim vi avviserà con questa linea aggiuntiva:

```
più RECENTE dello swap file! ~
```

Talora la linea

[non leggibile] ~

può apparire sotto il nome dello swap file. Questo può essere positivo o negativo, a seconda del contesto.

È positivo se una precedente sessione di modifica si è bloccata senza aver fatto alcuna modifica al file. In questo caso una lista directory dello swap file mostrerà che è lungo zero byte. Potete cancellarlo e andare avanti.

Va un po' male se non siete autorizzati a leggere lo swap file. Potete provare a visualizzare il file in sola-lettura, o abbandonare le modifiche. Su sistemi multiutente, se avete fatto voi le ultime modifiche usando un differente nome utente, una chiusura di sessione, seguita da una nuova sessione con l'altro nome utente potrebbe rimediare l'"errore di lettura". Oppure potreste individuare l'utente che ha fatto l'ultima modifica al file (o lo sta modificando adesso) e parlargli...

Va molto male se invece c'è un errore fisico di lettura sul disco che contiene lo swap file. Fortunatamente, questo non capita quasi mai. Potete visualizzare il file in sola-lettura (se ci riuscite), per vedere l'entità delle modifiche che erano state "dimenticate". Se siete responsabili della modifica di quel file, siate pronti a ridigitare le vostre ultime modifiche.

CHE FARE?

\*swap-exists-choices\*

Se sono supportate le scelte di dialogo, vi verrà chiesto di selezionare una di queste sei opzioni:

Swap file ".main.c.swp" già esistente! ~

[O] Apri sola-lettura, (E) Apri comunque, (R)ecupera, (Q) Esci, (A)nnulla, (D) Cancellalo: ~

- O Aprire il file in sola lettura. Usate questa opzione se volete solo esaminare il file ma non recuperarlo. Potete fare questa scelta quando sapete che qualcun altro sta modificando il file, e volete solo leggerlo senza fare modifiche.
- E Modificare comunque il file. Usatela con prudenza! Se un'altra sessione di Vim stesse modificando il file, potreste trovarvi con due versioni differenti dello stesso file. Vim cercherà di avvisarvi quando ciò accade, ma è meglio essere sicuri piuttosto che doversi pentire.
- R Recuperare il file dal file di swap. Usate questa opzione se sapete che il file di swap contiene modifiche che volete recuperare.
- Q Esci. Questo evita di modificare il file. Usate questa opzione se c'è un'altra sessione che sta modificando lo stesso file.  
Se avete appena avviato Vim, ciò chiuderà la sessione. Se avete avviato Vim con più file in diverse finestre, chiuderà la sessione solo se esiste un file di swap per il primo. Usando un comando di modifica, il file non verrà caricato e verrete riportati al file che stavate modificando.
- A Annulla. Come Q [Esci], ma non esegue eventuali comandi ulteriori. Utile eseguendo uno script che modifica parecchi file, ad es. una sessione con molte finestre.
- D Cancellare il file di swap. Usate questa opzione se siete sicuri di non averne più bisogno. Ad esempio se non contiene modifiche, o quando il file è più nuovo del file di swap.  
In Unix questa scelta è mostrata solo se il processo che ha creato il file di swap non risulta più attivo.

Se non appaiono le scelte di dialogo (state usando una versione di Vim che non le supporta) dovrete fare tutto manualmente. Per recuperare il file, usate questo comando: >

:recover

Non sempre Vim riesce a scoprire che esiste già un file di swap per un dato file. Questo avviene quando l'altra sessione di modifica mette il file di swap in una directory diversa, o se il percorso del file è differente quando è modificato su computer diversi. Quindi non fidatevi sempre degli avvisi di Vim.

Se veramente non voleste vedere questo messaggio, potreste aggiungere il flag "A" all'opzione '[shortness](#)'. Ma sarebbe veramente strano se ne aveste bisogno.

Per saperne di più riguardo alla cifratura e al file di swap, vedere [|:recover-crypt|](#).

---

#### \*11.4\* Altre letture

|                               |  |
|-------------------------------|--|
| <a href="#"> swap-file </a>   | Spiega dove viene creato il file di swap e qual è il suo nome.   |
| <a href="#"> :preserve </a>   | Salvare manualmente il file di swap sul disco.   |
| <a href="#"> :swapname </a>   | Vedi il nome del file di swap per il file corrente.  |
| <a href="#">'updatecount'</a> | Numero di caratteri immessi dopo i quali il file di swap viene salvato sul disco.                          |
| <a href="#">'updatetime'</a>  | Tempo dopo il quale il file di swap viene salvato sul disco.   |
| <a href="#">'swapsync'</a>    | Se i buffer in memoria vengono scaricati su disco ("synced") ogni volta che il file di swap viene salvato. |
| <a href="#">'directory'</a>   | Elenco delle directory dove viene memorizzato il file di swap.   |
| <a href="#">'maxmem'</a>      | Limite di utilizzo della memoria prima che il file di swap venga salvato.                                  |
| <a href="#">'maxmemtot'</a>   | Come sopra ma per tutti i file aperti.   |

---

Capitolo seguente: [|usr\\_12.txt|](#) Trucchi ingegnosi

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 [azc100](#) at gmail dot com