usr_45.txt Per Vim version 7.4. Ultima modifica: 2008 Nov 15

VIM USER MANUAL - di Bram Moolenaar
Traduzione di questo capitolo: Paolo Giovannelli

Selezionate la vostra lingua

I messaggi in Vim possono essere dati in molte lingue. Questo capitolo spiega come cambiare quale venga usata. Così vengono illustrati i diversi modi per lavorare con file in varie lingue.

Capitolo seguente: |usr_90.txt| Installare Vim
Capitolo precedente: |usr_44.txt| Evidenziazione della vostra sintassi
Indice: |usr_toc.txt|

==========

45.1 Lingua per i messaggi

Quando avviate Vim, esso controlla l'ambiente di lavoro per trovare quale lingua stiate utilizzando. Generalmente ciò funziona bene ed ottenete dei messaggi nella vostra lingua (se sono disponibili). Per vedere quale sia la lingua corrente, usate questo comando: >

```
:language
```

Se risponde con "C", ciò significa che viene impiegato il default, ovvero l'inglese.

> Nota:
> L'utilizzo di lingue diverse è possibile solo se Vim era stato compilato appositamente per gestirlo. Per scoprire se funziona, utilizzate il comando ":version" e verificate l'output per "+gettext" e "+multi_lang". Se ci sono, siete a posto. Altrimenti, se vedete "-gettext" o "-multi_lang" dovrete cercare un altro Vim.

Cosa fare se voleste i vostri messaggi in una lingua diversa? Ci sono diversi modi. Quale potreste usare dipende dalle capacità del vostro sistema.
Il primo modo consiste nell'impostare l'ambiente di lavoro per la lingua desiderata prima di avviare VIM. Esempio per Unix: >

```
env LANG=de_DE.ISO_8859-1  vim
```

Ciò funziona solamente se la lingua è presente nel vostro sistema. Il vantaggio è che tutti i messaggi della GUI e gli oggetti nelle librerie useranno anch'essi la lingua giusta. Uno svantaggio è che dovete farlo prima di avviare VIM. Se invece volete cambiare la lingua mentre Vim è in esecuzione, potete usare il secondo metodo: >

```
:language fr_FR.ISO_8859-1
```

In questo modo potete provare diversi nomi per la vostra lingua. Otterrete un messaggio di errore quando tale funzione non fosse supportata dal vostro sistema. Non ottenete alcun messaggio d'errore quando i messaggi tradotti non siano disponibili. VIM automaticamente ritornerà ad usare l'inglese.
Per sapere quali lingue sono supportate cercate la cartella ove esse sono elencate. Nel mio sistema, ad esempio, è "/usr/share/locale". Su alcuni sistemi "/usr/lib/locale". La pagina del manuale per "setlocale" potrebbe darvi un'indicazione su dove essa si trovi nel vostro sistema.
State attenti a digitare il nome esattamente come potrebbe essere. Le maiuscole e le minuscole sono importanti, ed i caratteri '-' e '_' vengono facilmente confusi.

Potete anche impostare una lingua diversa per i messaggi, il testo in lavorazione ed il formato della data. Vedere |:language|.

REALIZZARE DA SOLI LA TRADUZIONE DEI MESSAGGI

Se i messaggi tradotti non fossero disponibili nella vostra lingua, potreste scriverli voi stessi. Per far ciò, procuratevi il codice sorgente di Vim ed il pacchetto gettext GNU. Dopo aver scompattato i sorgenti, potrete trovare le istruzioni nella cartella src/po/README.txt.

Non è particolarmente difficile fare la traduzione. Non avrete bisogno di essere programmatori. Dovrete semplicemente conoscere sia l'inglese che la lingua in cui state traducendo, ovviamente.
Quando sarete soddisfatti con la traduzione pensate a renderla disponibile ad altri. Caricatela su vim-online (**http://vim.sf.net**) o spedendo un mail al maintainer Vim <maintainer@vim.org> o entrambe le cose.

==========

45.2 Lingua per i menù

I menù di default sono in inglese. Per poter impiegare la vostra lingua locale devono essere tradotti. Normalmente ciò viene svolto automaticamente per voi se l'ambiente di lavoro è impostato per la vostra lingua, proprio come per i messaggi. Non sarà necessario fare di più per questo. Ma ciò funziona solo se le traduzioni nella vostra lingua sono disponibili.
Supponete di essere in Germania, di avere impostato la lingua tedesca e di preferire "file" a "Datei". Potreste ripassare ai menù in inglese in questo modo: >

```
:set langmenu=none
```

È anche possibile specificare una lingua: >

```
:set langmenu=nl_NL.ISO_8859-1
```

Come sopra la differenza tra "-" e "_" è rilevante. Ad ogni modo, la differenza tra le maiuscole e le minuscole è ignorata in questo contesto.
L'opzione **'langmenu'** deve essere impostata prima che i menù vengano caricati. Dopo aver definito i menù infatti, cambiare **'langmenu'** non ottiene alcun effetto. Comunque, inserite il comando per impostare **'langmenu'** nel vostro file "vimrc".
Se realmente volete cambiare la lingua dei menù mentre utilizzate Vim, potete farlo in questo modo: >

```
:source $VIMRUNTIME/delmenu.vim
:set langmenu=de_DE.ISO_8859-1
:source $VIMRUNTIME/menu.vim
```

C'è solo uno svantaggio: Tutti i menù definiti da voi se ne andranno. Dovrete perciò ridefinirli.

REALIZZARE DA SOLI LA TRADUZIONE DEI MENU'

Per sapere quali traduzioni dei menù sono disponibili, guardate in questa cartella:

```
$VIMRUNTIME/lang ~
```

I file sono denominati menù_**{linguaggio}**.vim. Se non vedete la lingua che intendete utilizzare, potete fare voi stessi le traduzioni. Il modo più semplice per farlo è copiare uno dei file lingua esistenti e modificarlo.
Prima trovate il nome della vostra lingua con il comando ":language". Usate questo nome, ma con tutte le lettere rese minuscole. Allora copiate il file nella vostra cartella di runtime, quella trovata prima in **'runtimepath'**. Ad esempio, in Unix sarebbe: >

```
:!cp $VIMRUNTIME/lang/menu_ko_kr.euckr.vim ~/.vim/lang/menu_nl_be.iso_8859-1
.vim
```

Troverete suggerimenti per la traduzione in "$VIMRUNTIME/lang/README.txt".

==========

45.3 Utilizzare un'altra codifica.

Vim osserva che i file che state per aprire sono codificati per la vostra lingua. Per molte lingue europee tale codifica è "latin1". Allora ogni byte

è un solo carattere. Ciò significa che sono possibili 256 differenti caratteri. Per le lingue asiatiche ciò non è sufficiente. Queste usano principalmente una codifica a doppio byte, che rende disponibili oltre 10000 possibili caratteri. Ciò non è ancora sufficiente quando un testo contenga molte lingue diverse. Qui è dove entra Unicode. È stato progettato per includere tutti i caratteri utilizzati nelle lingue comunemente usate. Questa è la "super codifica che sostituisce tutte la altre". Ma non è ancora molto utilizzata.

Fortunatamente, Vim supporta tre tipi di codifica. E, con alcune restrizioni, potete utilizzarle anche quando il vostro ambiente di lavoro usa un'altra lingua rispetto al testo.

Ciononostante, quando aprite solo dei file codificati nella vostra lingua, il default dovrebbe lavorare bene e non dovrete fare altro. Ciò che segue è importante solo quando lavorate con lingue differenti.

> Nota:
> L'utilizzo di molteplici codifiche funziona solo se Vim era stato compilato per gestirle. Per scoprire se ciò funziona, utilizzate il comando ":version" e cercate nell'output "+multi_byte". Se lo trovate significa che siete a posto. Se invece vedete "-multi_byte" significa che dovrete cercare un'altra versione di Vim.

USARE UNICODE NELLA GUI

Il bello di Unicode è che le altre codifiche possono essere convertite ad esso e viceversa senza perdere informazioni. Quando impostate Vim per utilizzare Unicode internamente, potrete lavorare con file in qualsiasi codifica.

Sfortunatamente, il numero di sistemi che supportano Unicode è ancora limitato. Così è improbabile che la vostra lingua lo utilizzi. Dovrete dire a Vim che volete usare Unicode, e come interfacciarsi con il resto del sistema.

Iniziamo con la versione GUI di Vim, che può usare i caratteri Unicode. Ciò potrebbe funzionare: >

```
:set encoding=utf-8
:set guifont=-misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
```

L'opzione **'encoding'** dice a Vim quale codifica di caratteri utilizzate. Ciò si applica al testo nei buffer (i file su cui state lavorando), ai registri, agli script di Vim, etc. Potete guardare **'encoding'** come un'impostazione per le funzioni interne di Vim.

Questo esempio presuppone che abbiate questo font installato nel vostro sistema. Il nome nell'esempio è per il sistema Xwindow. Questo font si trova in un pacchetto che viene utilizzato per migliorare xterm con il supporto Unicode. Se non avete questo font, potete trovarlo qui:

http://www.cl.cam.ac.uk/~mgk25/download/ucs-fonts.tar.gz ~

Per MS-Windows, alcuni font hanno un numero limitato di caratteri Unicode. Provate usando il font "Courier New". Potete usare il menù Edit/Select Font per selezionare e provare i font disponibili. Tuttavia solo i font con ampiezza fissa posso essere utilizzati. >

```
:set guifont=courier_new:h12
```

Se non dovesse funzionare, cercate di ottenere un fontpack. Se Microsoft non l'ha spostato, potete trovarlo qui:

http://www.microsoft.com/typography/fonts/default.aspx ~

Ora avete detto a Vim di usare Unicode internamente e di visualizzare il testo con un font Unicode. I caratteri che digitato arrivano ancora nella codifica originale del vostro originale. Ciò richiede di convertirli in Unicode. Dite a Vim il linguaggio da cui convertire con l'opzione **'termencoding'**. Potete farlo così: >

```
:let &termencoding = &encoding
:set encoding=utf-8
```

Ciò assegna il vecchio valore di **'encoding'** a **'termencoding'**, prima di impostare **'encoding'** ad utf-8. Dovrete controllare se questi comandi funzionano davvero con la vostra installazione. Potrebbe funzionare

particolarmente bene usando un metodo di input per una lingua asiatica, e volete scrivere del testo in Unicode.

USARE UNICODE IN UN TERMINALE UNICODE

Esistono terminali che supportano Unicode direttamente. Lo standard xterm che viene con Xfree è uno di essi. Usiamolo come esempio.
Prima di tutto, xterm deve essere compilato con il supporto ad Unicode. Guardate |UTF8-xterm| per sapere come controllare ciò e come compilarlo in caso di necessità.
Avviate xterm con l'argomento "-u8". Potreste anche avere bisogno di specificare un font. Esempio: >

```
xterm -u8 -fn -misc-fixed-medium-r-normal--18-120-100-100-c-90-iso10646-1
```

Ora potete avviare Vim entro questo terminale. Impostate 'encoding' ad 'utf-8' come prima. E questo è tutto.

USARE UNICODE IN UN TERMINALE NORMALE

Supponiamo che vogliate lavorare con dei file Unicode, ma non abbiate un terminale con lo supporti. Potete fare ciò con Vim, tuttalpiù i caratteri non supportati dal terminale non verranno visualizzati. L'aspetto del testo verrà preservato. >

```
:let &termencoding = &encoding
:set encoding=utf-8
```

Questo è lo stesso che era stato usato per la GUI. Ma funziona diversamente: Vim convertirà il testo visualizzato prima di inviarlo al terminale. Ciò evita che il display visualizzi strani caratteri.
Affinché ciò funzioni è necessario che la conversione tra 'termencoding' ed 'encoding' sia possibile. Vim convertirà da latin1 ad Unicode, cosicché funzioni sempre. Per altre conversioni è richiesta la funzionalità |+iconv|.
Provate a modificare un file con caratteri Unicode. Noterete che Vim mette un punto interrogativo (oppure una sottolineatura o qualche altro carattere) nei posti dove c'è un carattere che il terminale non può visualizzare. Muovete il cursore sino al punto interrogativo ed usate questo comando: >

```
ga
```

Vim visualizzerà una linea con il codice del carattere. Ciò vi dà un'idea di che carattere sia. Potrete poi trovarlo in una tabella Unicode. Potreste realmente leggere un file in questo modo se aveste molto tempo a disposizione.

Nota:
Poiché 'encoding' è usato per tutto il testo entro Vim, modificarlo invaliderebbe tutto il testo non ASCII. Potete notare ciò usando i registri ed il file 'viminfo' (ad esempio, un modello di ricerca memorizzato). Si raccomanda perciò di impostare 'encoding' nel vostro file vimrc, e lasciarlo stare.

==
45.4 Elaborare file con una codifica differente

Supponete di avere impostato Vim per l'uso di Unicode e che vogliate modificare un file Unicode a 16 bit. Sembra semplice, giusto? Bene, Vim in realtà usa internamente la codifica utf-8, perciò la codifica a 16 bit deve essere convertita, poiché c'è una differenza tra il set di caratteri (Unicode) e la codifica (utf-8 o 16 bit).
Vim cercherà di individuare il tipo di file con il quale state lavorando. Esso usa nomi di codifica situati nell'opzione 'fileencodings'. Quando si usa Unicode, il valore standard è:"ucs-bom,utf-8,latin1". Ciò significa che Vim controlla il file per vedere se contiene una di queste codifiche:

ucs-bom Il file deve iniziare con un 'Byte Order Mark (BOM)'. Ciò consente la rilevazione delle codifiche unicode a 16 bit, 32 a bit ed utf-8

utf-8 utf-8 unicode. È rifiutata quando una sequenza di byte è illegale in utf-8.

latin1 La vecchia codifica ad 8-bit. Funziona sempre.

Quando iniziate a lavorare su un file Unicode a 16 bit, nel quale è presente un BOM, Vim individua ciò e converte il file in utf-8 quando lo legge. L'opzione **'fileencoding'** (senza la s finale) è impostata al valore individuato. In questo caso è "utf-16le". Ciò significa Unicode, due byte e un 'little-endian'. Questo formato di file è comune in MS-Windows, (ad esempio per i file di registro).

Quando si scrive un file, Vim confronta **'fileencoding'** con **'encoding'**. Se sono differenti, il testo sarà convertito.

Un valore nullo in **'fileencoding'** significa che nessuna conversione deve essere eseguita.

Se il valore di **'fileencodings'** di default non andasse bene per voi, allora impostatelo per le codifiche che volete che Vim tenti. Solo quando un valore venisse trovato non valido verrebbe usato quello successivo. Impostare "latin1" come primo non funziona, perché non è mai illegale. Un esempio, per tornare al giapponese quando il file non ha un BOM e non è utf-8: >

```
:set fileencodings=ucs-bom,utf-8,sjis
```

Vedere |**encoding-values**| per i valori suggeriti. Altri valori potrebbero funzionare altrettanto bene. Ciò dipende dal tipo di conversione disponibile.

FORZARE UNA CODIFICA

Se il rilevamento automatico non funzionasse, dovrete dire a Vim quale codifica sia usata dal file. Esempio: >

```
:edit ++enc=koi8-r russian.txt
```

La parte "++enc" specifica il nome della codifica da usare solamente per questo file. Vim convertirà il file dalla codifica specificata, il russo in questo esempio, in **'encoding'**. **'fileencoding'** verrà così impostato per la codifica specificata, cosicché la conversione inversa può essere fatta quando si scrive il file.

Lo stesso argomento può essere utilizzato scrivendo un file. In tal modo potete realmente usare Vim per convertire un file. Esempio: >

```
:write ++enc=utf-8 russian.txt
```

<

Nota:
La conversione può causare una perdita di caratteri. Convertire da una codifica qualsiasi in Unicode e viceversa è generalmente libera da questi problemi, a meno che ci siano caratteri illegali. La conversione da Unicode in altre codifiche spesso perde informazioni quando ci sia più di un linguaggio nel file.

==========

45.5 Impostare la lingua del testo

Le tastiere dei computer non hanno più di un centinaio di tasti. Alcune lingue hanno un centinaio di caratteri, Unicode ne ha più di 100000. Così come potete digitare questi caratteri?

Prima di tutto, quando non usate troppi caratteri speciali, potete usare i digrami. Questo argomento è già stato spiegato in |**24.9**|.

Quando utilizzate una lingua che usa molti più caratteri di quanti ne abbia la vostra tastiera, vorrete usare un Input Method (IM). Ciò richiede di imparare la traduzione dai tasti digitati ai caratteri risultanti. Quando vi serve un IM, ne avrete probabilmente già uno installato nel vostro sistema. Potrebbe funzionare con Vim come con altri programmi. Per i dettagli vedere |**mbyte-XIM**| per il sistema XWindow e |**mbyte-IME**| per MS-Windows.

KEYMAP

Per alcune lingue, il set di caratteri è diverso da quello latino, ma impiega un numero simile di caratteri. È perciò possibile mappare i tasti ai

caratteri. Vim utilizza le 'keymap' per questo.
Supponete di voler scrivere in ebraico. Potete caricare la keymap così: >

```
:set keymap=hebrew
```

Vim cercherà di trovare un file di keymap per voi. Ciò dipende dal valore di "encoding". Se nessun file corrispondente venisse trovato, otterreste un messaggio di errore.

Ora potete digitare in ebraico in Insert mode. In Normal mode e quando digitate un comando ":", Vim automaticamente passerà all'inglese. Potete usare questo comando per passare dall'ebraico all'inglese: >

```
CTRL-^
```

Questo funziona solo in Insert mode e in Comand-line mode. In Normal mode fa qualcosa di completamente diverso (salta ad un altro file).
L'uso della keymap è indicato nel messaggio della modalità, se avete l'opzione 'showmode' attivata. Nella gui Vim indicherà l'uso delle keymap con un colore diverso del cursore.
Potete anche cambiare l'uso della keymap con le opzioni 'iminsert' e 'imsearch'.

Per vedere la lista delle mappature, usate questo comando : >

```
:lmap
```

Per trovare quali file di keymap siano disponibili, nella GUI potete usare il menù Edit/Keymap. Altrimenti potete usare questo comando: >

```
:echo globpath(&rtp, "keymap/*.vim")
```

CREARE DA SOLI LE KEYMAP

Potete creare da soli i vostri file di keymap. Non è particolarmente difficile. Iniziate con un file di keymap simile a quello della lingua che volete impiegare. Copiatelo nella cartella "keymap" nella vostra directory di runtime. Ad esempio, in Unix potreste usare la cartella "~/.vim/keymap".
Il nome del file di keymap deve assomigliare a questo:

```
keymap/{nome}.vim ~
```

oppure

```
keymap/{nome}_{encoding}.vim ~
```

{nome} è il nome della keymap. Scegliete un nome che sia ovvio, ma differente dalle keymap esistenti (a meno che non vogliate sostituire un file di keymap esistente).
{nome} non può contenere un underscore. Volendo, aggiungete la codifica usata dopo un underscore. Esempi:

```
keymap/hebrew.vim ~
keymap/hebrew_utf-8.vim ~
```

I contenuti del file dovrebbero spiegarsi da sé. Guardate qualcuna delle keymap distribuite con VIM. Per i dettagli, vedere |mbyte-keymap|.

ULTIMA RISORSA

Se tutti gli altri metodi fallissero, potrete comunque inserire qualsiasi carattere con **CTRL-V**:

encoding	type	range
8-bit	CTRL-V 123	decimal 0-255
8-bit	CTRL-V x a1	hexadecimal 00-ff
16-bit	CTRL-V u 013b	hexadecimal 0000-ffff
31-bit	CTRL-V U 001303a4	hexadecimal 00000000-7fffffff

Non digitate gli spazi. Vedere |i_CTRL-V_digit| per i dettagli.

==========

Capitolo seguente: |**usr_90.txt**| Installare Vim

  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