

FIM : Fbi IMproved

dezperado _ presso _ autistici.org

20 Maggio 2007

Un *image viewer* nello spirito di Vim

MOHP 2007, Francavilla Al Mare

```
svn export http://code.autistici.org/svn/fim
```

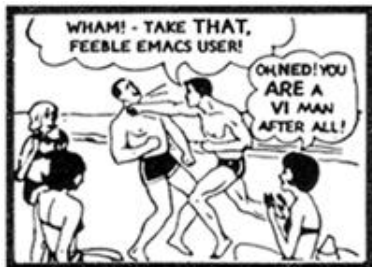


The Visual text editor - the Vi choice

- ▶ Chi non ha mai usato Vi su Unix ?
- ▶ Chi non ci ha mai combattuto all'inizio ?

The Visual text editor - the Vi choice

- ▶ Chi non ha mai usato Vi su Unix ?
- ▶ Chi non ci ha mai combattuto all'inizio ?
- ▶ C'è chi lo ama!



- ▶ C'è chi si batterebbe per Vi!

invarianti mondo UNIX

- ▶ pochi tasti fondamentali per usare Vi

Inoltre utilities `less`, `more`, con `vi` hanno in comune molti tasti:
'h', 'j', 'k', 'l', ' ', '/', ':', 'q', 'n'...

invarianti mondo UNIX

- ▶ pochi tasti fondamentali per usare Vi
- ▶ ..ostici al novizio

Inoltre utilities `less`, `more`, con `vi` hanno in comune molti tasti:
'h', 'j', 'k', 'l', ' ', '/', ':', ':q', ':n'...

invarianti mondo UNIX

- ▶ pochi tasti fondamentali per usare Vi
- ▶ ..ostici al novizio
- ▶ ..ma mnemonici alla lunga!

Inoltre utilities `less`, `more`, con `vi` hanno in comune molti tasti:
'h', 'j', 'k', 'l', ' ', '/', ':', 'q', 'n'...

Vi vs Vim

Vi su BSD (Bill Joy, 1976+) :

- ▶ nato per essere usato su un modem a 300 baud!

Vim su Unix (Bram Moolenaar, 1992+)

E .. per le immagini ? Esiste un *power image viewer* ?

Vi vs Vim

Vi su BSD (Bill Joy, 1976+) :

- ▶ nato per essere usato su un modem a 300 baud!
- ▶ interfaccia minimale

Vim su Unix (Bram Moolenaar, 1992+)

E .. per le immagini ? Esiste un *power image viewer* ?

Vi vs Vim

Vi su BSD (Bill Joy, 1976+) :

- ▶ nato per essere usato su un modem a 300 baud!
- ▶ interfaccia minimale
- ▶ svelto nell'uso

Vim su Unix (Bram Moolenaar, 1992+)

E .. per le immagini ? Esiste un *power image viewer* ?

Vi vs Vim

Vi su BSD (Bill Joy, 1976+) :

- ▶ nato per essere usato su un modem a 300 baud!
- ▶ interfaccia minimale
- ▶ svelto nell'uso

Vim su Unix (Bram Moolenaar, 1992+)

- ▶ un Vi 'pompato'

E .. per le immagini ? Esiste un *power image viewer* ?

Vi vs Vim

Vi su BSD (Bill Joy, 1976+) :

- ▶ nato per essere usato su un modem a 300 baud!
- ▶ interfaccia minimale
- ▶ svelto nell'uso

Vim su Unix (Bram Moolenaar, 1992+)

- ▶ un Vi 'pompato'
- ▶ ora Vim è un *power editor*, paragonabile a Emacs

E .. per le immagini ? Esiste un *power image viewer* ?

Fbi (Gerd Knorr 1999+)

- ▶ gira nel framebuffer di Linux
- ▶ è leggero

Siamo sicuri sia abbastanza ?

Fbi (Gerd Knorr 1999+)

- ▶ gira nel framebuffer di Linux
- ▶ è leggero
- ▶ supporta diversi formati grafici
- ▶ ha un wrapper per vedere file Adobe Postscript, PDF

Siamo sicuri sia abbastanza ?

Fbi (Gerd Knorr 1999+)

- ▶ gira nel framebuffer di Linux
- ▶ è leggero
- ▶ supporta diversi formati grafici
- ▶ ha un wrapper per vedere file Adobe Postscript, PDF
- ▶ layout dei tasti intuitivi

Siamo sicuri sia abbastanza ?

Fbi (Gerd Knorr 1999+)

- ▶ gira nel framebuffer di Linux
- ▶ è leggero
- ▶ supporta diversi formati grafici
- ▶ ha un wrapper per vedere file Adobe Postscript, PDF
- ▶ layout dei tasti intuitivi
- ▶ è un prodotto tedesco :)

Siamo sicuri sia abbastanza ?

Si potrebbe fare di meglio!

- ▶ Fbi non è configurabile :

Quindi ? Esiste un viewer che abbia tutte queste feature ?

Si potrebbe fare di meglio!

- ▶ Fbi non è configurabile :
- ▶ manca rebinding dei tasti

Quindi ? Esiste un viewer che abbia tutte queste feature ?

Si potrebbe fare di meglio!

- ▶ Fbi non è configurabile :
- ▶ manca rebinding dei tasti
- ▶ mancano autowidth, autoheight, scaling 'condizionale'

Quindi ? Esiste un viewer che abbia tutte queste feature ?

Si potrebbe fare di meglio!

- ▶ Fbi non è configurabile :
- ▶ manca rebinding dei tasti
- ▶ mancano autowidth, autoheight, scaling 'condizionale'
- ▶ ... → *scriptabilità*!?

Quindi ? Esiste un viewer che abbia tutte queste feature ?

Fim : Un *image viewer* modale!

Modaità :

- ▶ modalità *interattiva*

Feature di base

Fim : Un *image viewer* modale!

Modaità :

- ▶ modalità *interattiva*
- ▶ modalità *readline* (con libreria GNU readline)

Feature di base

Fim : Un *image viewer* modale!

Modaità :

- ▶ modalità *interattiva*
- ▶ modalità *readline* (con libreria GNU readline)

Feature di base

- ▶ rebinding completo tasti [a-zA-Z0-9...] con Control (modalità *interattiva*)

Fim : Un *image viewer* modale!

Modaità :

- ▶ modalità *interattiva*
- ▶ modalità *readline* (con libreria GNU readline)

Feature di base

- ▶ rebinding completo tasti [a-zA-Z0-9...] con Control (modalità *interattiva*)
- ▶ autowidth, autoheight da tastiera

Fim : Un *image viewer* modale!

Modaità :

- ▶ modalità *interattiva*
- ▶ modalità *readline* (con libreria GNU readline)

Feature di base

- ▶ rebinding completo tasti [a-zA-Z0-9...] con Control (modalità *interattiva*)
- ▶ autowidth, autoheight da tastiera
- ▶ comandi interni, variabili interne (in *readline*)

Fim : Un *image viewer* modale!

Modaità :

- ▶ modalità *interattiva*
- ▶ modalità *readline* (con libreria GNU *readline*)

Feature di base

- ▶ rebinding completo tasti [a-zA-Z0-9...] con Control (modalità *interattiva*)
- ▶ autowidth, autoheight da tastiera
- ▶ comandi interni, variabili interne (in *readline*)
- ▶ GNU *readline* interface con Tab autocompletion
- ▶ alias e interprete *if-while*

Comando goto

- ▶ :1 porta alla prima immagine

Comando goto

- ▶ :1 porta alla prima immagine
- ▶ :\$ porta all'ultima immagine in lista

Comando goto

- ▶ :1 porta alla prima immagine
- ▶ :\$ porta all'ultima immagine in lista
- ▶ :goto "\$" analogo

next, prev

- ▶ :1 porta alla prima immagine
- ▶ :\$ porta all'ultima immagine in lista
- ▶ :goto "\$" analogo
- ▶ :next porta alla prossima immagine

next, prev

- ▶ `:1` porta alla prima immagine
- ▶ `:$` porta all'ultima immagine in lista
- ▶ `:goto "$` analogo
- ▶ `:next` porta alla prossima immagine
- ▶ `:2next` porta due immagini avanti (sì, come in Vim)

next, prev

- ▶ `:1` porta alla prima immagine
- ▶ `:$` porta all'ultima immagine in lista
- ▶ `:goto "$` analogo
- ▶ `:next` porta alla prossima immagine
- ▶ `:2next` porta due immagini avanti (sì, come in Vim)
- ▶ `:next;prev;` porta avanti e quindi indietro

alias e bind

- ▶ `:alias 'n' 'next'`
- ▶ `:bind 'n' 'n'`
- ▶ ora si può premere 'n' in modalità interattiva per vedere la prossima immagine

alias e bind

- ▶ `:alias 'n' 'next'`
- ▶ `:bind 'n' 'n'`
- ▶ ora si può premere 'n' in modalità interattiva per vedere la prossima immagine
- ▶ `:bind 'N' '10n'`
- ▶ e 'N' per andare avanti di 10!

esempio di while

- ▶ `alias "endless_slideshow"`
`"while(1){next;display;sleep '1';}";`
- ▶ `alias "pornview" "echo 'press any key repeatedly to terminate' ;endless_slideshow";`
- ▶ `bind "C-p" "pornview";`

In realtà il comando `next`, è minimale, nel senso che non implica il `display` dell'immagine, ma solo l'avanzamento *dell'immagine corrente* in lista.

Ciò assicura un'adeguata configurabilità.

Perchè ?

... perchè sta all'utente di Fim decidere *come* usarlo: l'importante è che abbia la libertà di usarlo (smanettabilità, *meccanismi*, non parlando d' altro..)

ciò grazie al meccanismo degli *autocomandi*

Autocomandi ed eventi

Un comando che scatta all'occorrenza di un *evento*:

```
▶ autocmd "PostNext" "" "reload;"
```

è un *autocomando*!

L'invocazione di `next` fa scattare `reload` !

Autocomandi ed eventi

Un comando che scatta all'occorrenza di un *evento*:

- ▶ `autocmd "PostNext" "" "reload;"`
- ▶ `autocmd "PostReload" "" "{redisplay;}";`

è un *autocomando*!

L'invocazione di `next` fa scattare `reload` !

Posix regular expressions

```
▶ autocmd "PostDisplay" ".*png" "echo 'this is a  
png file'";
```

In questo caso, il comando echo scatta solo sui file png!

Regex search

`/. *png` fa scattare una 'ricerca' di file `.png`
`C-n` è un next dedicato al prossimo filename *che 'matcha'*

Sommario feature

- ▶ tab autocompletion
- ▶ possibilità' di script file
- ▶ costrutti while, if, variabili interne
- ▶ autocomandi con attivazione per filename
- ▶ comandi pre e post esecuzione da command line
- ▶ espressioni regolari sui filename per ricerca
- ▶ recording azioni :
start_recording, stop_recording, repeat_last
- ▶ interazione con sistema : `bind 'C-g' "system 'fbgrab fim.png'";`
- ▶ standard input/output ...
- ▶ etc...

Fim e' una patch applicata a Fbi

È stato programmato in C/C++ con STL (Standard Template Library), integrando Flex(Lex) e Bison(Yacc) e la libreria GNU readline al codice sorgente originale di Fbi.

Architettura un pò più ordinata che in Fbi, orientata (a grandi linee) verso 'Model-View-Controller'.

chi forse *vorrebbe* usarlo:

- ▶ chi ha Linux con il framebuffer device abilitato
- ▶ l'utente incallito di Vim, o il fan delle shell

chi forse *vorrebbe* usarlo:

- ▶ chi ha Linux con il framebuffer device abilitato
- ▶ l'utente incallito di Vim, o il fan delle shell
- ▶ chi non sopporta il mouse

chi forse *vorrebbe* usarlo:

- ▶ chi ha Linux con il framebuffer device abilitato
- ▶ l'utente incallito di Vim, o il fan delle shell
- ▶ chi non sopporta il mouse
- ▶ chi non sopporta o non usa X e le finestre

chi forse *vorrebbe* usarlo:

- ▶ chi ha Linux con il framebuffer device abilitato
- ▶ l'utente incallito di Vim, o il fan delle shell
- ▶ chi non sopporta il mouse
- ▶ chi non sopporta o non usa X e le finestre
- ▶ lo smanettone *imagofilo*

chi forse *potrebbe* aiutare nello sviluppo di Fim

- ▶ un visionario che vorrebbe integrare altre sorgenti di immagini in Fim

chi forse *potrebbe* aiutare nello sviluppo di Fim

- ▶ un visionario che vorrebbe integrare altre sorgenti di immagini in Fim
- ▶ chi abbia a cuore *l'ergonomicità* del software per *nerds*

chi forse *potrebbe* aiutare nello sviluppo di Fim

- ▶ un visionario che vorrebbe integrare altre sorgenti di immagini in Fim
- ▶ chi abbia a cuore *l'ergonomicità* del software per *nerds*
- ▶ ...chi mi inventa una macchina del tempo...

chi forse *potrebbe* aiutare nello sviluppo di Fim

- ▶ un visionario che vorrebbe integrare altre sorgenti di immagini in Fim
- ▶ chi abbia a cuore *l'ergonomicità* del software per *nerds*
- ▶ ...chi mi inventa una macchina del tempo...
- ▶ oppure...

idee che frullano nel cervello..

- ▶ finestrelle annidate, come in Vim o Irssi o Emacs..

idee che frullano nel cervello..

- ▶ finestrelle annidate, come in Vim o Irssi o Emacs..
- ▶ strutturare il linguaggio nella maniera più *intuitiva*

science fiction

- ▶ finestrelle annidate, come in Vim o Irssi o Emacs..
- ▶ strutturare il linguaggio nella maniera più *intuitiva*
- ▶ integrazione con *speech recognizer*

science fiction

- ▶ finestrelle annidate, come in Vim o Irssi o Emacs..
- ▶ strutturare il linguaggio nella maniera più *intuitiva*
- ▶ integrazione con *speech recognizer*
- ▶ integrazione con codice di analisi/ricerca di immagini

science fiction

- ▶ finestrelle annidate, come in Vim o Irssi o Emacs..
- ▶ strutturare il linguaggio nella maniera più *intuitiva*
- ▶ integrazione con *speech recognizer*
- ▶ integrazione con codice di analisi/ricerca di immagini
- ▶ integrabilità con qualche linguaggio esterno (Perl, Python), per la creazione di *moduli*

fine

Spero sia stata una presentazione interessante!

Domande ?

Referenze :

- ▶ `'dict "vi"'`
- ▶ `'dict "vim"'`
- ▶ `'man "readline"'`
- ▶ `'man "fbi"'`
- ▶ `'man "fim"'`
- ▶ `svn export http://code.autistici.org/svn/fim`
- ▶ `http://code.autistici.org/svn/fim/doc/FIM.TXT`
- ▶ `http://www.autistici.org/dezperado/fim/FIM.html`