

FIM: Fbi IMproved, v.0.5

A swiss army knife for viewing images

Michele Martone

Opensource Treffen München, 26.02.2016

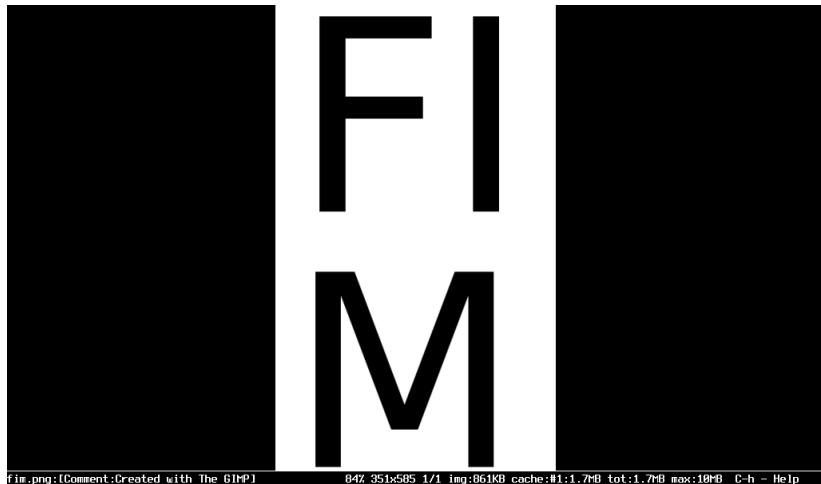
<http://www.nongnu.org/fbi-improved>

what ?

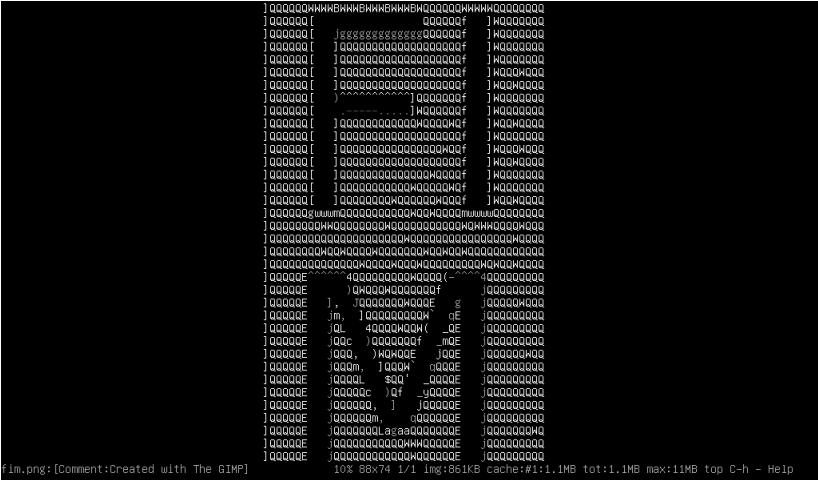
fm is an image viewer

- ▶ a modal one: two modes:
 - ▶ *normal* (using `Keystrokes` bound to `:commands`)
 - ▶ *command line* (using its `:scripting language`)
- ▶ scriptable:
 - ▶ *internally* (with its `:scripting language`)
 - ▶ *externally* (`$` from the shell)
- ▶ multiple graphical modes, e.g.:
 - ▶ SDL-1.2 (*Simple Directmedia Layer* for Xorg,...)
 - ▶ Linux framebuffer device; e.g. `/dev/fb0`
 - ▶ aalib (ascii art rendering, on a textual terminal)

a screenshot (either framebuffer or X/SDL)



an ascii art (aalib) rendering screenshot



why the name *FIM: Fbi IMproved* ?

- ▶ started as a fork of `fbi-1.31` (by Gerd Hoffmann)
- ▶ tribute to `vim` and `fbi`

why at all ?

an inspiring quote (pertaining the mutt mail user agent):

“All mail clients suck. This one just sucks less.”

Michael Elkins, circa 1995

how to invoke/use it ?

if you don't know, try:

- ▶ `$ fim --help` *#provides a help message*
- ▶ `$ fim --help=d` *#... more descriptive*
- ▶ `$ man fim` *#manual page for the 'fim' command*
- ▶ `$ man fimrc` *#... for its internal language and configuration*
- ▶ `C-h - Help` *# (the internal help)*

how to invoke, simply ?

the way you expect it:

- ▶ `$ fim file.jpg` *#opens a file*
- ▶ `$ fim file1.jpg -- --file2.jpg` *#opens two JPEG files*
- ▶ `$ fim .` *#opens all the files in the directory*

how to use it simply ?

the way you expect it to:

- ▶ `PageUp` *next*
- ▶ `PageDown` *previous*
- ▶ `-` *shrink (zoom out)*
- ▶ `+` *magnify (zoom in)*
- ▶ `Esc` *quit*

how to use it cleverly ?

with all the fingers on the alphabetic keys:

- ▶ **n** *go to the next file*
- ▶ **p** *go to the previous file*
- ▶ **q** *quit*
- ▶ **'** *jump back and forth*
- ▶ ...

a topical example ?

- ▶ **h** *pan to the left*
- ▶ **j** *pan down*
- ▶ **k** *pan up*
- ▶ **l** *pan to the right*

do we know these keys from elsewhere ?

how to script it ?

- ▶ `:` *enter readline mode*
- ▶ `:help` *command for online documentation*
- ▶ `:next` *go to the next file*
- ▶ `:prev` *go to the previous file*
- ▶ `:bind` *show/modify `key` bindings*
- ▶ `:` `Enter` *quit the readline mode*

alternative invocation modes ?

▶ `$ fim -R /media/pictures/` *#recursive directory traversal*

▶

`$ fim --background-recursive /media/pictures/` *#same, allowing viewing files while scanning for further ones¹*

¹New feature: still in trunk.

alternative invocation modes ?

flexible standard input use:

- ▶ `$ find /media -iname *.tiff | fim -` *#open a list of files*
- ▶ `$ vgrabbj -d /dev/video0 | fim -i` *#open an 'anonymous' file from the pipe*
- ▶ `$ cat cmds.fim | fim -p` *#execute commands by reading them from stdin*

alternative invocation modes ?

mark dump file names to stdout:

- ▶ `$ fim * > selection.txt` *#save list of selected filenames*
- ▶ `$ cat selection.txt | fim - > top-ten.txt` *#refine a selection*

why to script ?

access to very specific actions:

- ▶ `:help "goto"` requests "goto" help
- ▶ `:goto "2"` jump to the second file
- ▶ `:goto "+2"` go forward two files
- ▶ `:goto "-2"` step two files backwards
- ▶ `:goto "^"` go to the first file
- ▶ `:goto "$"` go to the last file
- ▶ `:goto "33%"` go to the file positioned around one third in the files list
- ▶ `:goto "/*flower*.jpg/"` jump to the first file matching this regular expression
- ▶ `:goto "+//"` jump to the next file matching the last inserted regular expression

but scripting is boring!

there are shortcuts footnoteLike the Up-down motions in vim.:

- ▶ `:2` *go to the second file*
- ▶ `:^` *go to the first file*
- ▶ `:$` *go to the last file*
- ▶ `/*.flower*.jpg` *regexp-based jump*

even shorten shortcuts ?

prepend number to key to repeat

- ▶ **2n** *go to next, twice*
- ▶ **3+** *magnify, three times*
- ▶ **4.** *repeat last action, four times*

image scaling ?

examples

- ▶ `:scale "20%"` *scale to 20%*
- ▶ `:scale "+20%"` *magnify by one fifth*
- ▶ `:scale "-20%"` *shrink by one fifth*
- ▶ `:scale "0.2"` *shrink to one fifth*
- ▶ `:scale "w"` *scale to width*
- ▶ `:scale "h"` *scale to height*
- ▶ `:scale "a"` *scale automatically*

keys ?

binding commands to keys:

- ▶ `:bind "n" "next;"` `n` will execute next
- ▶ `:bind "N" "10next;"` `N` will execute 10next
- ▶ `:bind "C-n" "goto '+//'"` *jump to next regexp match*

language elements

- ▶ `:alias "ne" "next"` *"ne" will do as "next"*
- ▶ `:os=scale;` *variable assignment*

current image variables, loops

- ▶ `:echo i:width` *certain fim variables are accessible*
- ▶ `:echo i:*` *display all of an image's variables*
- ▶ `:if(i:width>_screen_width){scale "w"}` *conditional execution*
- ▶ `:while(i:width>_screen_width){goto "+1"}` *iterating a cycle*

startup

- ▶ configuration:

- ▶ `$HOME/.fimrc`

- ▶ `/etc/fimrc`

- ▶ command history:

- ▶ `$HOME/.fim_history`

integration with other software

- ▶ with mailcap+mutt:

```
$ echo "image/*; fim %s" > $HOME/.mailcap #
```

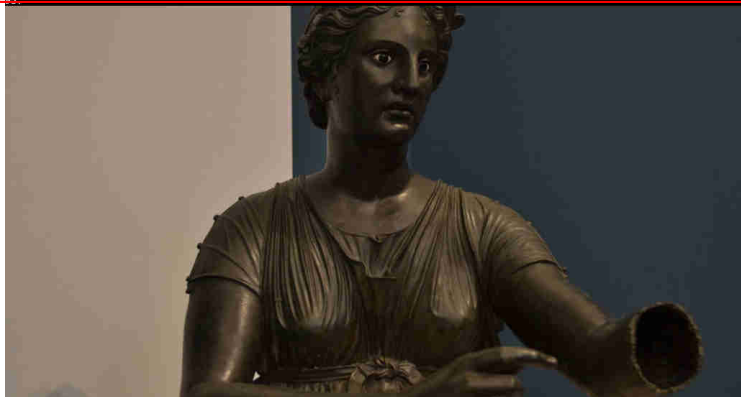
- ▶ multi-page files rendering (fimgs):

```
$ fimgs archive.rar #uncompress and display
```

- ▶ with midnight commander (mc), or elinks, ...

customization of **status line**, **overlay caption**, window caption

Busto di Diana. Da Pompei, 1817, rinvenuto presso i portici occidentali del santuario di Apollo. Bronzo, II s.a.C. MANN, inv. 48



sample.jpg [1/15 sec.] [f/4.0] [150400]

96% 1229x820 1/1 img:3.1MB cache:11.7.0MB tot:7.0MB C-h - Help

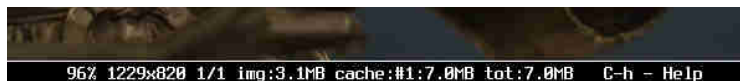
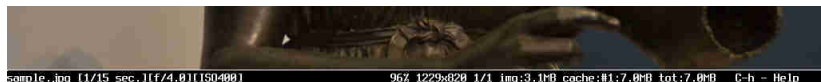
example fimrc: caption over image

Busto di Diana. Da Pompei, 1817, rinvenuto presso i portici occidentali del santuario di Apollo. Bronzo, II s.a.C. MANN, inv. 4895.



```
1 _caption_over_image=1 # display i:_comment
```

example fimrc: status line format specifier variables



```
1 # printf-like expandos
2 _info_fmt_str=
3 "%p%% %wx%h%L %i/%l%P img:%M cache:%C tot:%T";
```



```
4 _display_status_fmt= # EXIF tags as i:variables
5 "%N%?EXIF_ExposureTime?[%:EXIF_ExposureTime:]?".
6 "%?EXIF_FNumber?[%:EXIF_FNumber:]?".
7 "%?EXIF_ISOSpeedRatings?[ISO%:EXIF_ISOSpeedRatings:]?";
```

the image descriptions file: basics

```
vim --load-image-descriptions-file pets.dsc /petpics/*
```

```
1 # This is file dogs.dsc
2 DSC_0001.JPG a big dog
3 DSC_0002.JPG a big dog
4 DSC_0003.JPG a big dog on the couch
5 DSC_0004.JPG a big cat and a big dog
6 DSC_0005.JPG a fat dog
```

- ▶ `:goto "/couch/"` *jump to DSC_0003.JPG*
- ▶ `/couch` *jump to DSC_0003.JPG*
- ▶ `:limit "couch"` *limit browsable list to DSC_0003.JPG*

the image descriptions file: shebang syntax

```
1 DSC_0001.JPG a big dog
2 DSC_0002.JPG a big dog
3 DSC_0003.JPG a big dog on the couch
4 DSC_0004.JPG a cat and a big dog
5 DSC_0005.JPG a fat dog
```

can be expressed also via:

```
1 DSC_0001.JPG a big dog
2 DSC_0002.JPG #!fim:=
3 DSC_0003.JPG #!fim:+ on the couch
4 DSC_0004.JPG #!fim:^a cat and
5 DSC_0005.JPG #!fim:s/big/fat
```

this assigns the `:i:_comment` property of each file

the image descriptions file: **group variables set** and filtering

Assign the `:i:city` property to **several** images.

```
1 #!fim:city=Rome
2 DSC_0005.JPG Colosseum
3 #!fim:city=Naples
4 DSC_0006.JPG Bust of Diana
5 DSC_0007.JPG Vesuvius
6 #!fim:city= # unset the i:city property
7 ...
```

`:limit "city" "Naples"` *Will restrict the list to
DSC_0006.JPG, DSC_0007.JPG*

example fimrc: custom file deletion command

```
1 # not really recommended
2 bind "Del"    " f=i:_filename;"
3              ." list 'remove' f; f='';"
4              ." system 'rm' '-v' f;"
5              ." reload;"
```

Del *Pressing 'Delete' will delete the file as well*

example fimrc: custom file resize command

```
1 # useful for e.g. web publishing...
2 bind 'r' "o=i:_filename; n='halved_'.o;"
3         " system 'convert' '-resize' '50%' o n;"
4         " list 'push' n; goto '/''.n.'/' ;";
```

r *Pressing 'r' will create a resized file copy.*

example fimrc: use of `:autocmd`

```
1 # this is a built-in rule
2 autocmd "PostInteractiveCommand"
3     ".*DSC.*"
4     "if(_want_prefetch){prefetch;}";
```

why and when to use it ?

- ▶ its unique features set
- ▶ ergonomic/taste reasons, e.g. with mutt, pine
- ▶ minimalistic/embedded systems, e.g. Raspberry Pi or no X
- ▶ richer, screen-tolerating alternative to fbi

the end: thanks for your attention

<http://www.nongnu.org/fbi-improved>

extra: further and/or experimental features

- ▶ image prefetching + caching (faster loading)
- ▶ mip maps (faster scaling)
- ▶ font scaling (for large resolution monitors)
- ▶ **DJVU, PDF, EPS/PS** decoding
- ▶ decoding of archives **ZIP, RAR, ISO, ...** (via `libarchive`)
- ▶ command line auto-completion
- ▶ command line history
- ▶ sparse matrices rendering (with `librsb`)
- ▶ can use ImageMagick's `convert` for unknown formats
- ▶ read a file at a *specified byte offset*
- ▶ raw bytes as pixels/bitmap rendering
- ▶ rendering of text

extra: internals

- ▶ \approx 40000 C++ lines (out of the \approx 2000 original Fbi's lines)
- ▶ a parser written with flex and GNU bison
- ▶ uses the GNU readline and history libraries (the same used by bash)
- ▶ C++11's `std::thread`, STL library
- ▶ uses `libpng`, `libtiff`, `libgif`, `libexif`, `libjasper`,
...

extra: how to install it ?

on Debian/Ubuntu *testing/unstable*:

```
$ sudo apt-get install fim # :-)
```

on other distros: just check...

extra: how to install it ?

on Debian/Ubuntu, from source:

```
1 $ sudo apt-get install subversion \  
2 automake autoconf libtool libreadline6-dev \  
3 libjpeg-dev libpng-dev libtiff-dev libgif-dev \  
4 libsdl1.2-dev libsdl-gfx1.2-dev libaa1-dev \  
5 libdjvulibre-dev  
6 $ svn export http://svn.savannah.nongnu.org/svn/fbi-improved/trunk fim  
7 $ cd fim  
8 $ ./configure --enable-sdl  
9 $ make  
10 $ sudo make install
```