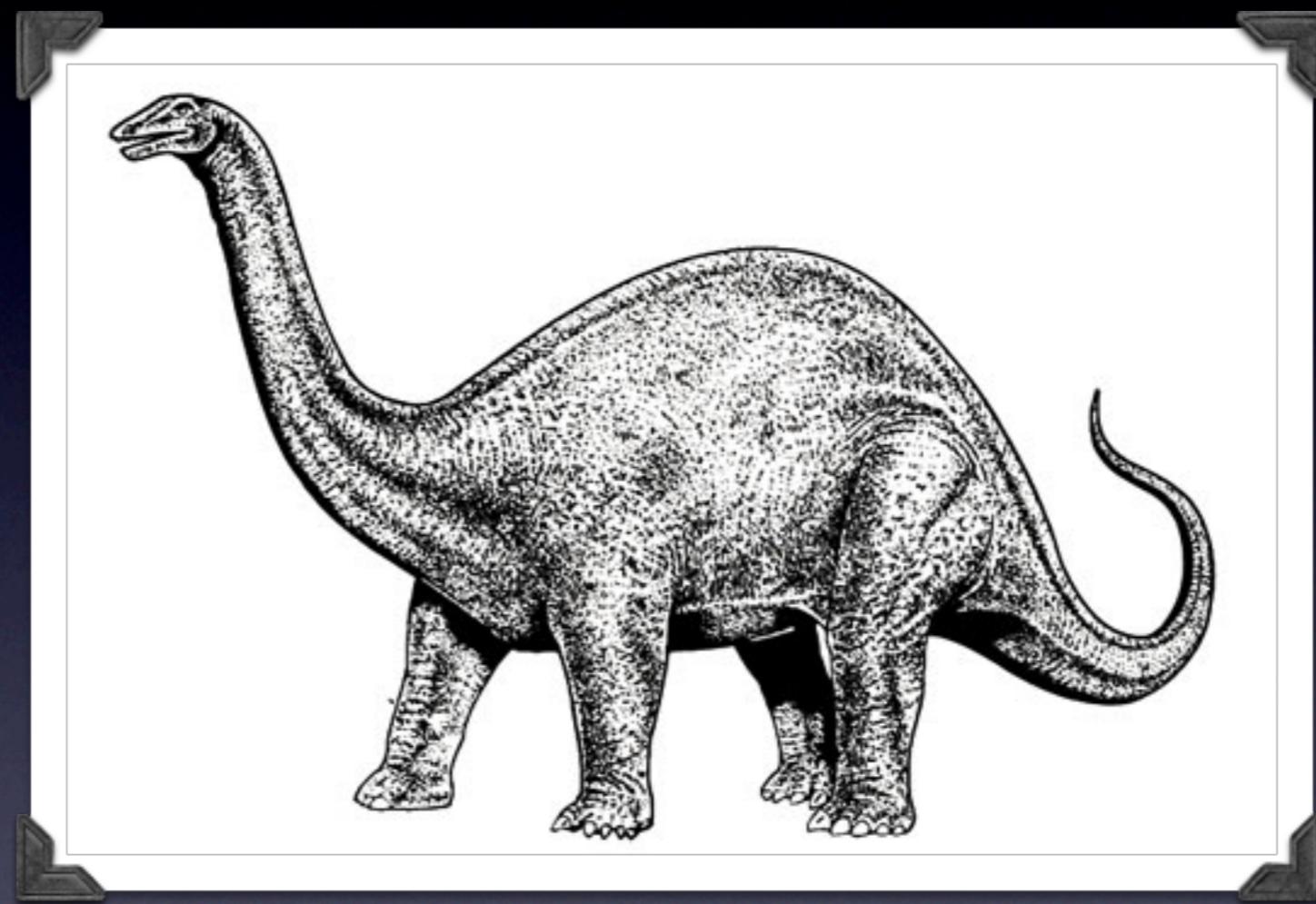
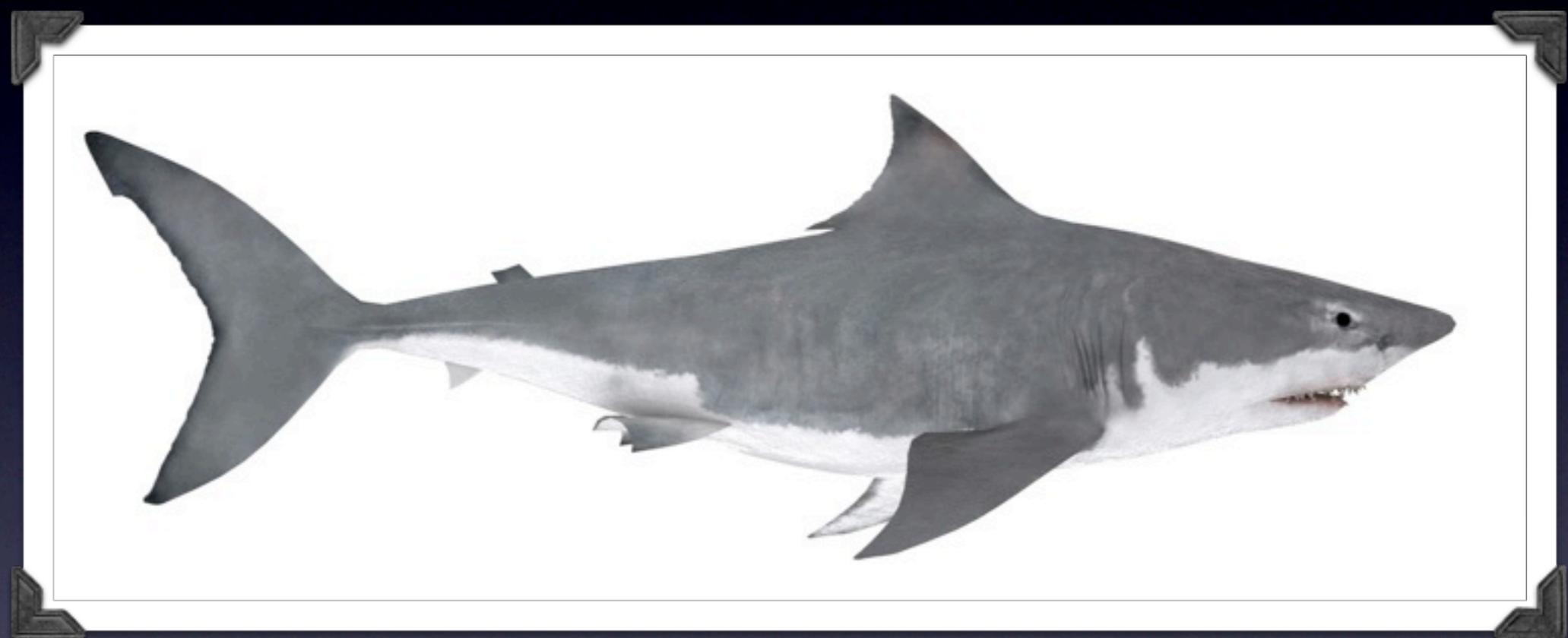


# Vim Fandango

Extended Edition – Director's Cut

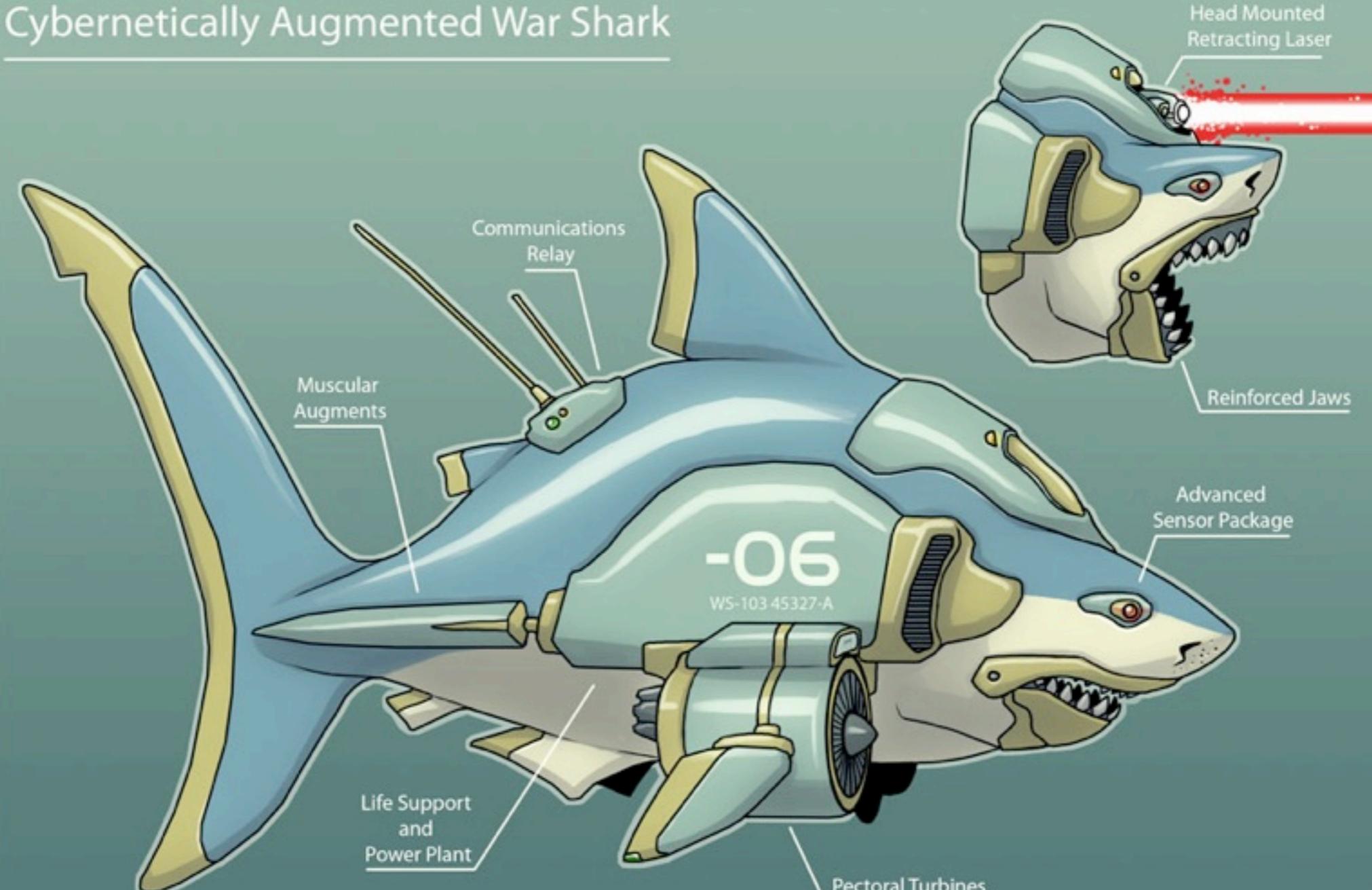
*Bill Odom  
St. Louis Perl Mongers*





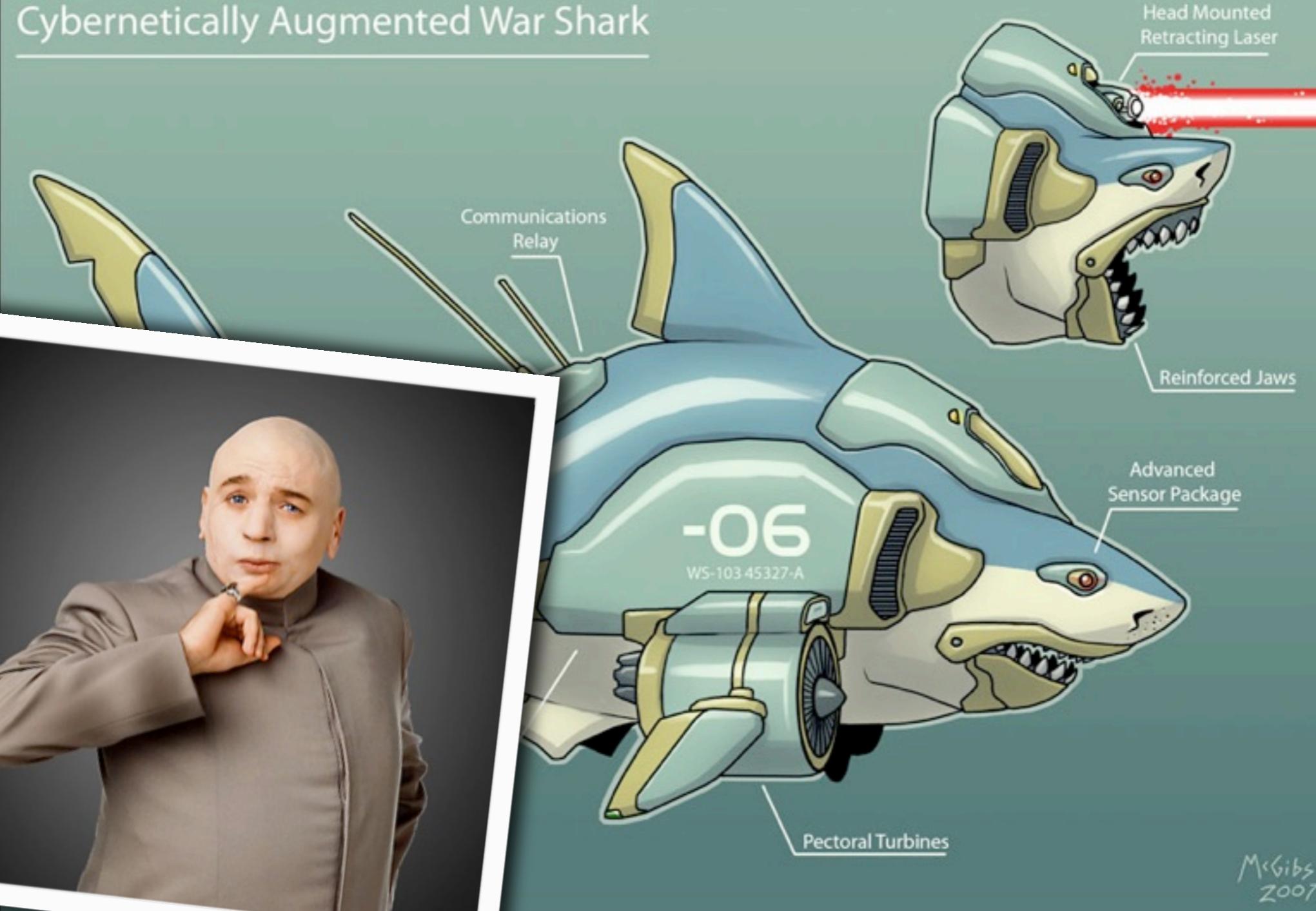
# WS-103

## Cybernetically Augmented War Shark



# WS-103

Cybernetically Augmented War Shark



# A Quick History Lesson

# What is Vim?

# Modal Text Editor





Based on Vi...

...written by Bill Joy...



...in 1976.

1976

So why bother?

# The Sales Pitch

# Multi-Language

## Terminal — bash — bodom — ttys000 — 180x55 — #1

```
$ ls
2html.vim      cdrdaoconf.vim    diff.vim      gedcom.vim    java.vim      messages.vim   pf.vim       rpcgen.vim   splint.vim   uil.vim
README.txt     cdrtoc.vim       dircolors.vim git.vim       javacc.vim   mgl.vim      pfmain.vim   rpl.vim      sup.vim     updatedb.vim
a2ps.vim       cf.vim          diva.vim      gitcommit.vim gitconfig.vim jess.vim      php.vim      rst.vim      spyce.vim  valgrind.vim
a65.vim        cfg.vim         django.vim   gitbase.vim   gitrebase.vim jgraph.vim   phtml.vim   rtf.vim      sql.vim    vb.vim
aap.vim        ch.vim          dns.vim      gitsendemail.vim gitproperties.vim jproperties.vim jsp.vim      ruby.vim   sqlanywhere.vim vera.vim
abap.vim       change.vim      docbk.vim   docbksgml.vim gkrellmrc.vim kconfig.vim  mma.vim      samba.vim  sqlforms.vim verilog.vim
abaqus.vim    changelog.vim   docbkxml.vim gnuplot.vim   gp.vim       kix.vim      modconf.vim plaintex.vim sas.vim      sqlinformix.vim verilogams.vim
abc.vim        chaskell.vim   dosbatch.vim dosini.vim   doxygen.vim  grads.vim   model.vim   pike.vim    sambavim  vgrindefsvim
abel.vim       cheetah.vim    dracula.vim groff.vim   dot.vim      gretl.vim   modula2.vim plm.vim      sash.vim   sclaracle.vim vhdl.vim
acedb.vim     chill.vim       dosini.vim   groovy.vim  dsl.vim      lace.vim   modula3.vim plp.vim      screen.vim  scheme.vim  vim.vim
ada.vim        chordpro.vim  doxygen.vim groovy.vim  group.vim   ld.vim      monk.vim   plssql.vim sd.vim      squid.vim  scilab.vim  viminfo.vim
aflex.vim     cl.vim          dracula.vim groovy.vim  ldapconf.vim ldif.vim     moo.vim     po.vim      sshconfig.vim virata.vim
ahdl.vim      clean.vim      dracula.vim groovy.vim  grub.vim   ldif.vim     mp.vim      postscr.vim sd.vim      st.vim     sshdconfig.vim vmasm.vim
alsaconf.vim  clipper.vim    dsl.vim      groovy.vim  haml.vim   lftp.vim     mplayerconf.vim pov.vim      stata.vim  sdc.vim     voscn.vim
amiga.vim    cmake.vim      dtm.vim      grub.vim   haml.vim   lhaskell.vim mrxvtrc.vim povini.vim sed.vim      stp.vim     strace.vim  vrmcl.vim
aml.vim       cmusrc.vim    dtm.vim      grub.vim   haml.vim   libao.vim   msmessages.vim ppwiz.vim   sendpr.vim sudoers.vim vsejcl.vim
ampl.vim     cobol.vim      dtrace.vim  grub.vim   haml.vim   haskell.vim msql.vim     prescribe.vim sgml.vim   services.vim syncolor.vim wget.vim
ant.vim       coco.vim      dylan.vim   gtkrc.vim  haml.vim   hamster.vim lifelines.vim mupad.vim   privoxy.vim sgmldecl.vim synload.vim whitespace.vim
antlr.vim    colortest.vim  dylanintr.vim haml.vim   haml.vim   haskell.vim lilo.vim     limits.vim  procmail.vim sgmlnx.vim sysctl.vim winbatch.vim
apache.vim   conaryrecipe.vim dylanlid.vim haml.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     muttrc.vim  progress.vim sh.vim      tads.vim    takcmp.vim  wml.vim
apachestyle.vim conf.vim      edif.vim     help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     mysql.vim   prolog.vim  sicad.vim   tags.vim    takout.vim  wsh.vim
arch.vim     config.vim    eiffel.vim   help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hex.vim     named.vim   protocols.vim simula.vim   sieve.vim   sindacmp.vim  wsmpl.vim
art.vim      context.vim   elf.vim      help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hercules.vim  hex.vim     logtalk.vim  natural.vim  purifylog.vim sinda.vim   sindaout.vim taskdata.vim xdefaults.vim
asm.vim       cpp.vim       elinks.vim   help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hercules.vim  hex.vim     loginaccess.vim  named.vim   protocols.vim  sisu.vim   taskedit.vim xinetd.vim
asm68k.vim   crm.vim      elinks.vim   help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hercules.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  skill.vim   tasm.vim    taskdata.vim xinetd.vim
asmh8300.vim crontab.vim   elmfilt.vim help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hercules.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slang.vim   tcl.vim    taskdata.vim xinetd.vim
asn.vim      cs.vim        erlang.vim  help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hercules.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slice.vim   tcl.vim    taskdata.vim xinetd.vim
aspperl.vim  csc.vim      eruby.vim   help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hercules.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slpconf.vim tex.vim    texinfo.vim texmf.vim   xmodmap.vim
aspvbs.vim   csh.vim      esmtpcr.vim help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slpreg.vim  tidy.vim   texinfo.vim xpm.vim
asterisk.vim csp.vim      esqlc.vim   help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slppspi.vim tf.vim    tilde.vim  texinfo.vim xpm2.vim
asteriskvm.vim css.vim      esterel.vim help.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slrnrcc.vim tli.vim   tli.vim    texinfo.vim xquery.vim
atlas.vim    cterm.vim    eterm.vim   htmlcheetah.vim haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  sm.vim    tpp.vim   tli.vim    texinfo.vim xquery.vim
autohotkey.vim ctrlh.vim   evviews.vim htmldjango.vim haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  smarty.vim  trasys.vim tli.vim    tpp.vim   tli.vim    texinfo.vim xquery.vim
autoit.vim   cuda.vim     exim.vim    html4.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  smcl.vim   trustees.vim tsalt.vim  xs.vim
automake.vim cupl.vim     expect.vim  htmllos.vim  haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  smil.vim   trusts.vim tsalt.vim  xs.vim
ave.vim      cuplsm.vim   exports.vim  ia64.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slice.vim   trustees.vim tsalt.vim  xs.vim
awk.vim      cvs.vim      fasm.vim    ibasic.vim  haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slpconf.vim tidy.vim   tilde.vim  trustees.vim z8a.vim
ayacc.vim   cvsrc.vim    fetchmail.vim icon.vim   haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slpreg.vim  tidy.vim   tilde.vim  trustees.vim z8a.vim
b.vim        cweb.vim     fgl.vim     idl.vim    haml.vim   hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slppspi.vim tf.vim    tidy.vim   tilde.vim  trustees.vim z8a.vim
baan.vim    cynlib.vim   flexwiki.vim idl.lang.vim indent.vim hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  slrnrcc.vim tli.vim   tli.vim    tilde.vim  trustees.vim z8a.vim
basic.vim   cynpp.vim    focexec.vim idl.lang.vim indent.vim hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  sm.vim    tpp.vim   tli.vim    tilde.vim  trustees.vim z8a.vim
bc.vim      d.vim        form.vim    forth.vim  initex.vim hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  smarty.vim  trasys.vim tli.vim   tpp.vim   tli.vim    tilde.vim  trustees.vim z8a.vim
bdf.vim     dcd.vim     forth.vim   fortran.vim initng.vim hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  smcl.vim   trustees.vim tsalt.vim  xs.vim
bib.vim     dcl.vim     forth.vim   fortran.vim initng.vim hastepreproc.vim haxe.vim     lisp.vim     hostconf.vim  hex.vim     logtalk.vim   natural.vim  purifylog.vim  smil.vim   trusts.vim tsalt.vim  xs.vim
bindzone.vim debchangelog.vim foopro.vim   framescript.vim ipfilter.vim mailaliases.vim manconf.vim pamconf.vim maple.vim papp.vim  reva.vim   snnsnet.vim tsscl.vim
blank.vim   debcontrol.vim foopro.vim   framescript.vim ipfilter.vim mailaliases.vim manconf.vim pamconf.vim maple.vim papp.vim  reva.vim   snnsnet.vim tsscl.vim
bst.vim     def.vim      freebasic.vim fstab.vim   iss.vim     master.vim  matlab.vim pcap.vim   rib.vim    snnsnet.vim tsscl.vim
btm.vim    denyhosts.vim fwmm.vim    fwmm2m4.vim jal.vim   matlab.vim  maxima.vim pdf.vim   rnc.vim   snnsnet.vim tsscl.vim
bzx.vim    desc.vim     fwmm.vim    fwmm2m4.vim jam.vim   matlab.vim  maxima.vim perl.vim  robots.vim snnsnet.vim tsscl.vim
calendar.vim desktop.vim   fwmm2m4.vim gdmo.vim   jargon.vim mel.vim
$
```

# Multi-Interface

" Press ? for help

.. (up a dir)  
 /Users/bodom/.env/.vim/  
 +after/  
 +colors/  
 +compiler/  
 +doc/  
 +ftplugin/  
 +plugin/  
 +ruby/  
 +syntax/  
 -diff\_with\_saved.vim  
 -filetype.vim  
 -insert\_matching\_spaces.vim  
 -map\_movement\_keys.vim  
 -map\_text\_autoformat\_keys.vim  
 -paste\_into\_c\_string.vim  
 -redir\_messages.vim  
 -search\_enhancements.vim  
 -search\_with\_skip.vim  
 -setAbbreviations.vim  
 -switch\_quotes.vim  
 -toggle\_highlights.vim

```

550 runtime set_abbreviations.vim
551
552
553 " Colors
554 "
555 colorscheme wnodom
556
557 "
558 "
559 " Extension settings (plugins, syntax fi>
560 "
561
562 " Perl syntax
563 "
564 let perl_include_pod = 1
565 " unlet perl_want_scope_in_variables
566 " unlet perl_extended_vars
567 let perl_string_as_statement = 1
568 " unlet perl_no_sync_on_sub
569 " unlet perl_no_sync_on_global_var
570 " let perl_sync_dist = 100
571 let perl_fold = 1
572 let perl_fold_blocks = 1
573 " let perl_nofold_packages = 1
574 " let perl_nofold_subs = 1
575
576 "
577 " NERD Tree
578 "
579 let NERDChristmasTree=0
580 let NERDTreeHighlightCursorline=1
581 let NERDTreeWinSize=48
582
583 "
584 " Taglist
585 "
586 " WIN32: Correct pathname for Windows.
587 "
588 if has("unix")
589   let Tlist_Ctags_Cmd = '/usr/local/bin/ctags'
590 elseif has("win32")
591   let Tlist_Ctags_Cmd = 'ctags.exe'
592 endif
593
594 let Tlist_Use_Right_Window = 1
595
596 "
597 "
598 "
599 " Functions
600 "
601
602 function! ConfigureWindow()

```

55 "
56 if exists('&macatsui')
57 set nomacatsui
58 endif
59 "
60 " Set the antialias, linespace, and guifont together, since different
61 " fonts look better with different settings. (These can be set separately,
62 " but it's just more convenient to keep them together like this.)
63 "
64 set antialias linespace=0 guifont=Inconsolata:h15
65 "set antialias linespace=0 guifont=Droid\_Sans\_Mono:h13
66 "set antialias linespace=0 guifont=Consolas:h15
67 "set antialias linespace=2 guifont=Monaco:h13
68 "set antialias guifont=Andale\_Mono:h14
69 "set noantialias guifont=Fixedsys\_True\_Type\_Font:h15
70 set lines=45 columns=100
71 "
72 " winpos doesn't (yet) work on MacVim.
73 "
74 "winpos 338 44
75 "
76 "
77 endif
78 "
79 "
80 " end \_gvimrc

~/gvimrc [R0] 80,1 Bot

\*help.txt\* For Vim version 7.2. Last change: 2008 Jul 21

VIM – main help file

Move around: Use the cursor keys, or "h" to go left, "l" to go right, "j" to go down, "k" to go up, "l" to go right. h l k j

Close this window: Use ":q<Enter>".

Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!).

Jump to a subject: Position the cursor on a tag (e.g. |bars|) and hit **CTRL-T**. With the mouse: ":set mouse=a" to enable the mouse (in xterm or GUI). Double-click the left mouse button on a tag, e.g. |bars|.

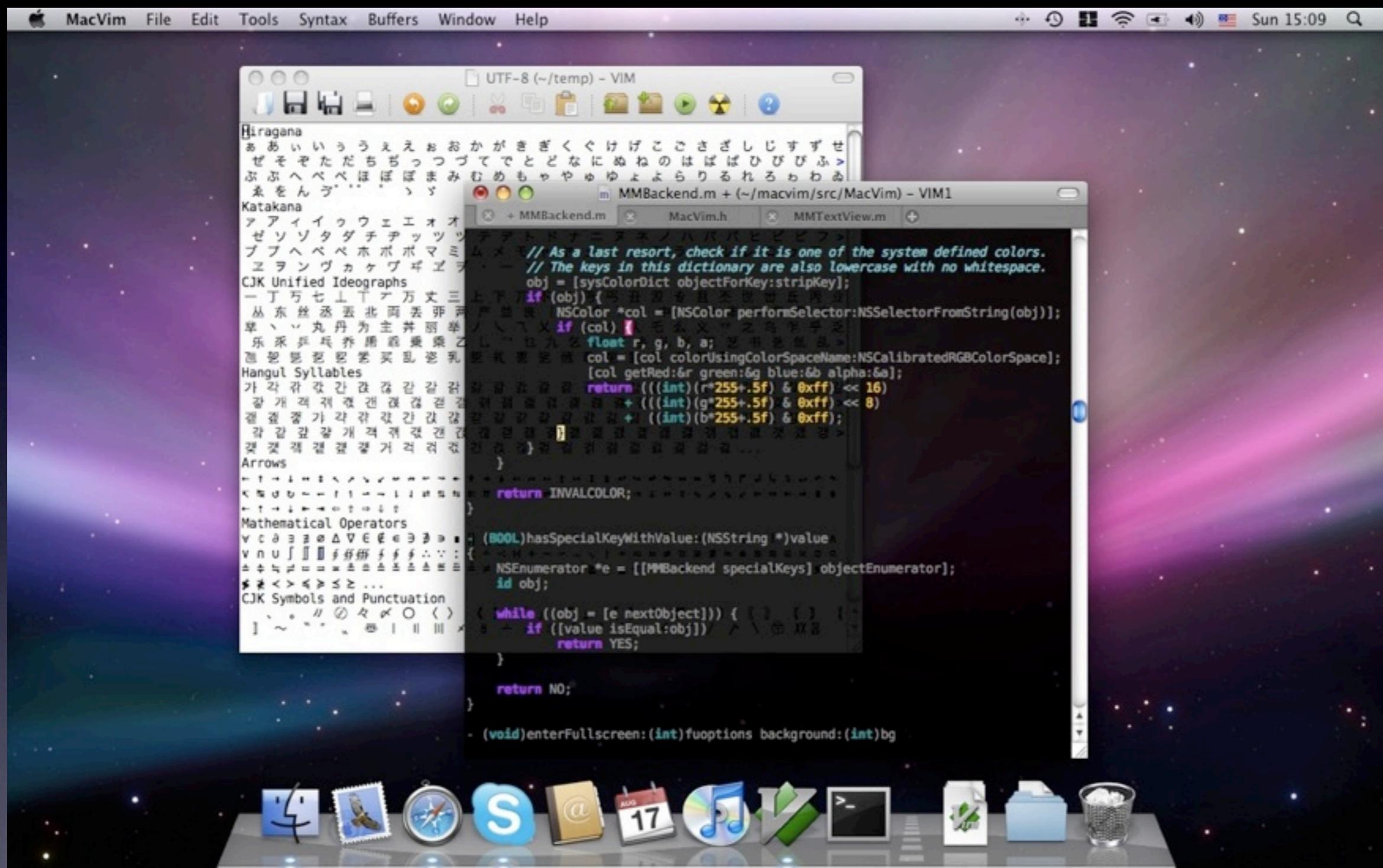
Jump back: Type **CTRL-O** (repeat to go further back).

Get specific help: It is possible to go directly to whatever you want help on, by giving an argument to the **:help** command. It is possible to further specify the context:

WHAT	PREPEND	EXAMPLE
Normal mode command	(nothing)	:help x
Visual mode command	v_	:help v_u
Insert mode command	i_	:help i_<Esc>
Command-line command	:	:help :quit
Command-line editing	c_	:help c_<Del>
Vim command argument	-	:help -r
Option	'	:help 'textwidth'

\*help-context\* 1,1 Top

~/gvimrc [R0] 576,0-1 95% help.txt [Help][R0]



# Multi-Platform

Session Edit View Bookmarks Settings Help

feedalizer.rb

```
1 require "rss/maker"
2 require "open-uri"
3 require "htree"
4 require "htree-fix"
5
6 class Feedalizer
7   attr_reader :source
8
9   def initialize(url, &b)
10    @source = parse_source(url)
11    @rss = RSS::Maker::R
12    feed.link = url # A
13    if block_given?
14      instance_eval &blo
15      output!
16    end
17  end
18
19  def feed
20    @rss.channel
21  end
22
23  def scrape_items(xpath)
24    @source.each(xpath_q
25      yield @rss.items.new_item, element
26    end
27  end
28
29  # TODO: Add Tidy backend
30  def parse_source(url)
31    open(url) { |io| HTTree(io).to_rexml.elements }
32  end
33
```

-- INSERT --

normal.c (~vim/vim6/src) - VIM

File Edit Tools Syntax Buffers Window Words Help

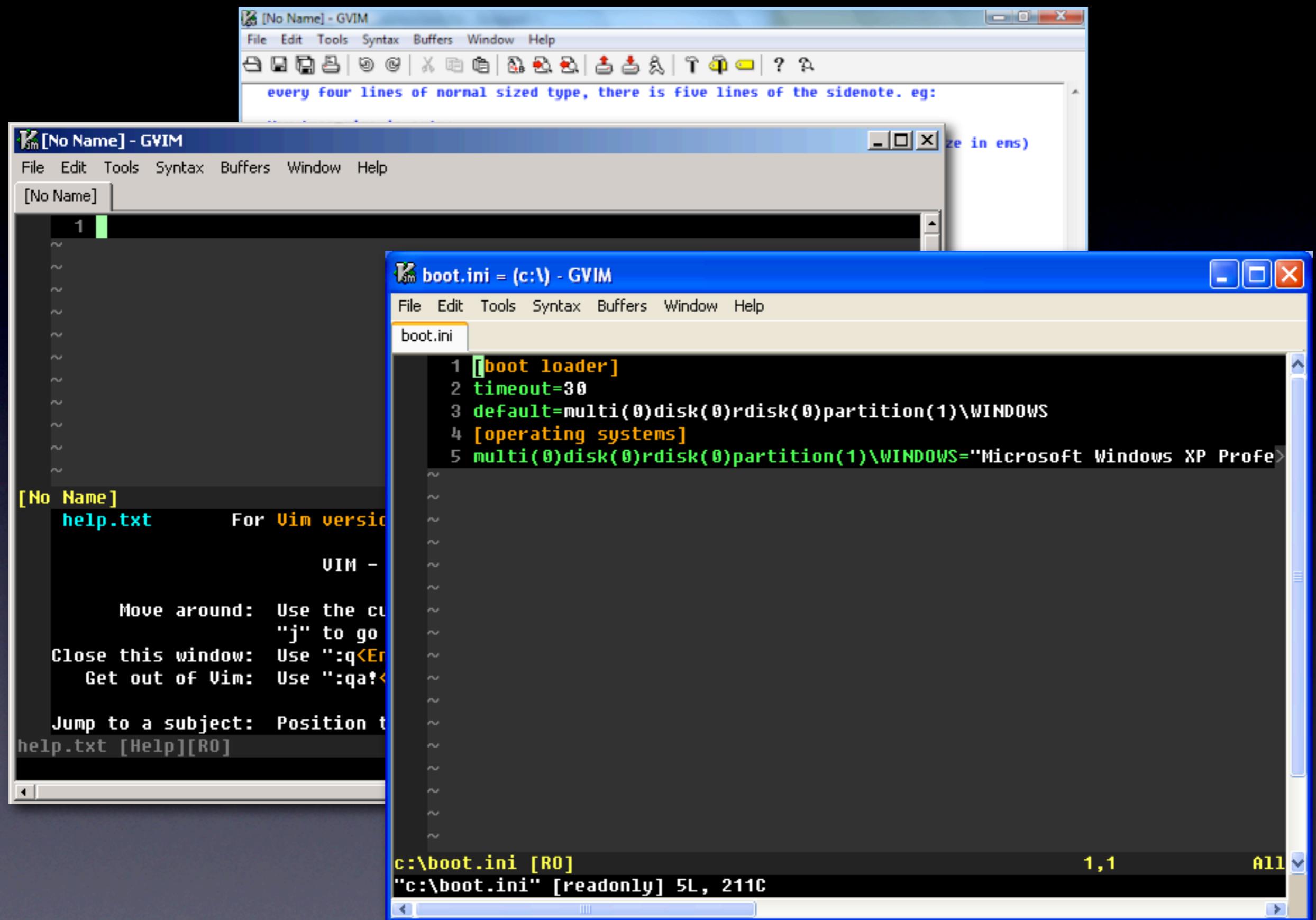
find\_start\_of\_word(pos)
 pos\_T \*pos;
 char\_u \*line;
 int cclass;
 int col;

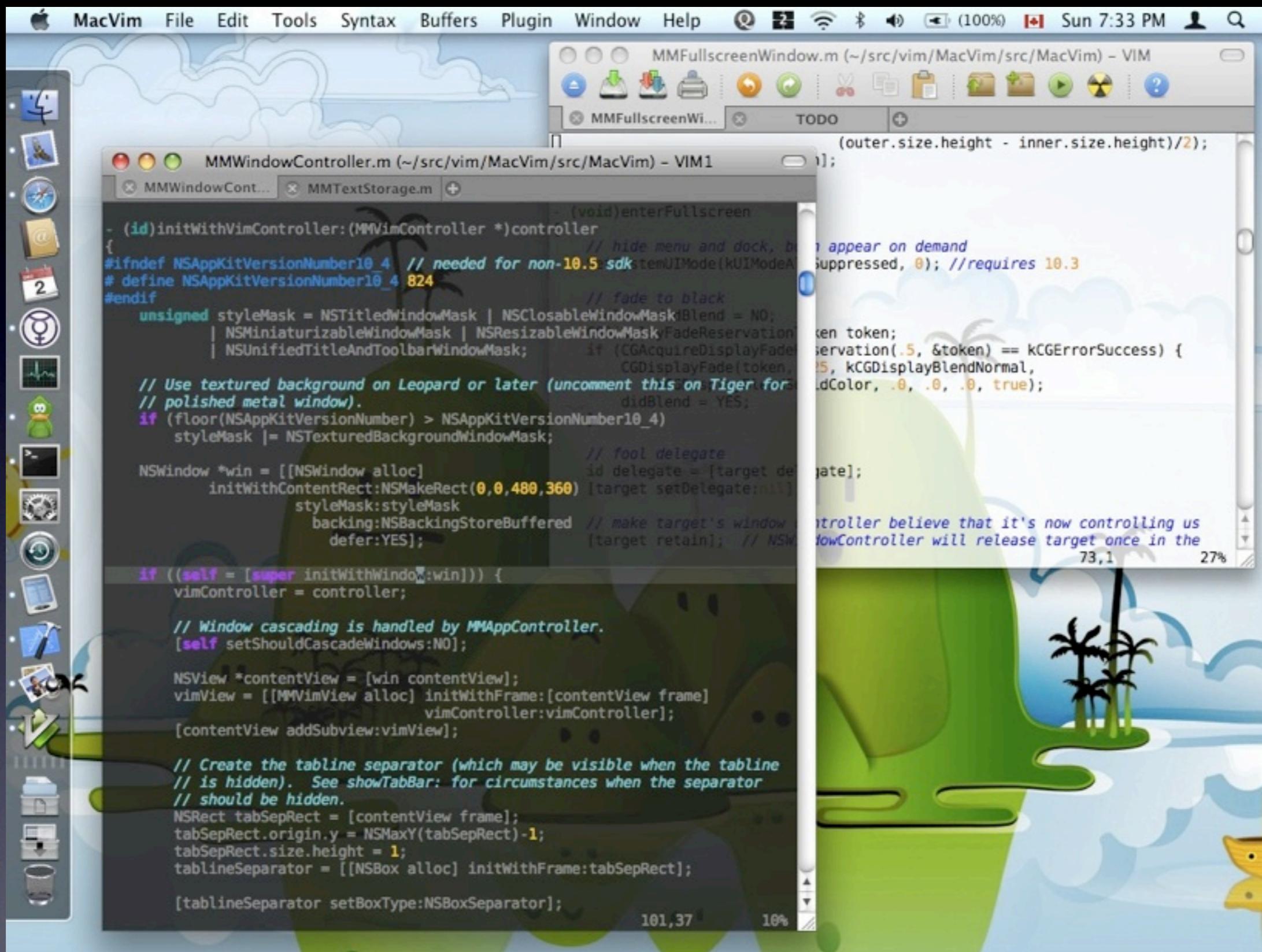
 line = ml\_get(pos->lnum);
 cclass = get\_mouse\_class(line + pos->col);

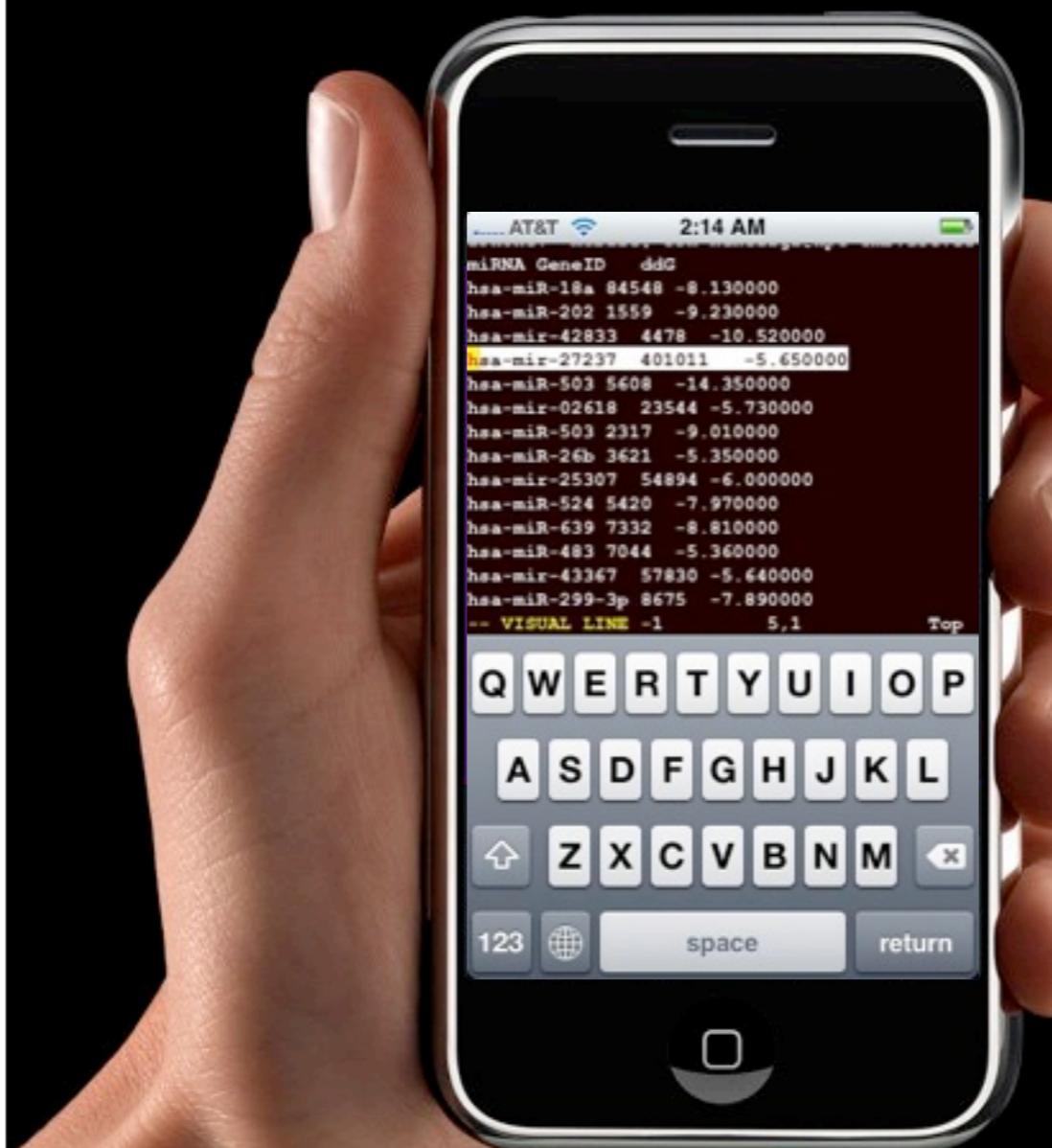
 while (pos->col > 0)
 {
 col = pos->col - 1;
 }

2689,19 34%

27,41 Top











MAGIC *Etch A Sketch*® SCREEN

Horizontal  
Ctrl

Sababa toys

©2004 Sababa, Inc. All Rights Reserved

Vertical  
Ctrl





```
#include <stdio.h>
#include <time.h>
int main ()
{
    time_t rawtime;
    struct tm *timeinfo;
    time(&rawtime);
    timeinfo = localtime (&rawtime);
    int day = timeinfo->tm_mday;
    int month = timeinfo->tm_mon + 1;
    int year = timeinfo->tm_year + 1900;
    int age = year - 1981;
    char *suffix;
    if (age % 10 == 1 && age != 11) {
        suffix = (char *) "st";
    } else if (age % 10 == 2 && age != 12) {
        suffix = (char *) "nd";
    } else if (age % 10 == 3 && age != 13) {
        suffix = (char *) "rd";
    } else {
        suffix = (char *) "th";
    }
    if (month == 8 && day == 16) {
        printf("Happy %d%s Birthday, Tim!\n", age,
               suffix);
    } else {
        printf("It's not your birthday..\n");
    }
    return 0;
}
```

# Active Community

welcome home : vim online

SPONSOR VOTE  
Vim development for features

V the editor

BUY HELP LEARN  
the Vim book Uganda Vim

not logged in ([login](#))

Google Custom Search

Search

Home Advanced search

About Vim Community News Sponsoring Trivia Documentation Download

Scripts Tips My Account

Site Help

What is Vim?

Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the vi editor distributed with most UNIX systems. Vim is

**News**

Vim 7.3 released!

[2010-08-15] Finally, after two years of development, Vim 7.3 is released! It includes an awful lot of bug fixes and new features. For the details see the [release notes](#). (Bram Moolenaar)

Vim 7.3f BETA version available

[2010-08-13] This is the last BETA release before the final release. Please report any problem you notice. If you have been using Vim 7.3, you have been! See the [announcement](#).

Vim 7.3e BETA version available

[2010-08-08] This might be the last BETA release. All known problems have been fixed, it is in the [announcement](#). (Bram Moolenaar)

[more news...](#) Show you like

Recent Script Updates

[2010-08-18] Smart Parentheses programming

Vim 7.3.003 is the current version

Ads by Google Python IDE Faster, Easier Python

Vim Tips Wiki:Community Portal – Vim Tips Wiki

Create a new wiki Wnodom My Home My talk MORE Log out

Project page Discussion

## Vim Tips Wiki:Community Portal [edit this page]

This is the starting point for all editors of the Vim Tips wiki.

- Please see our [policy introduction](#) for how-to information.
- To follow changes to this wiki, and to look for new events and problem edits, use [recent changes](#).
- To discuss a tip, please edit the tip page and add to the Comments at the bottom (see the [discussion guidelines](#) for why we avoid the discussion pages for tips).
- On other pages (for example, this page), you are welcome to edit the [discussion page](#) to talk about the content.
- If you want help editing, you could start with the [tutorial](#) on the Central Wikia, or see our [editing help](#).
- There are more help pages in the [help category](#).
- And even more on [wikimedia.org](#).

More about the Vim community can be found at [vim.org](#).

Contents [hide]

- 1 Asking questions
  - 1.1 Creating a screenshot
- 2 Following (watching) changes
- 3 Downloading tips
- 4 Mirror sites

## Asking questions [edit]

On this wiki, we discuss how to improve tips, but we don't discuss how to use Vim.

Questions about Vim should be sent to the [Vim mailing list](#). The newsgroup [comp.editors](#) also

Ridiculously  
Powerful

Ridiculously  
Configurable

# Open Source

Good for You

# Good for Orphans

:help uganda



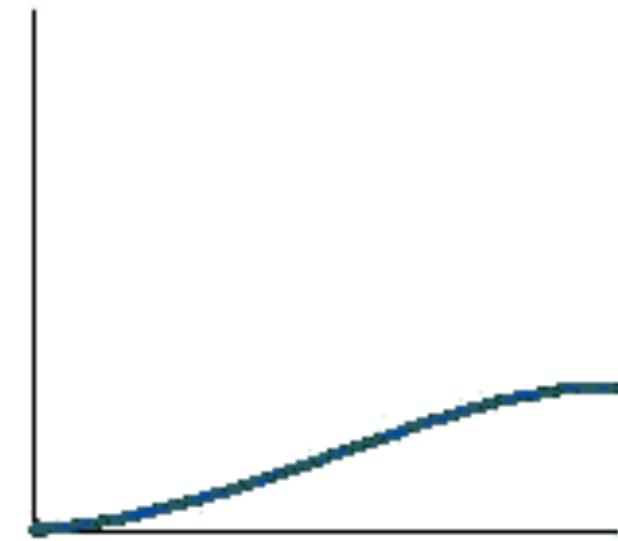
The Fine Print:

# Classical learning curves for some common editors

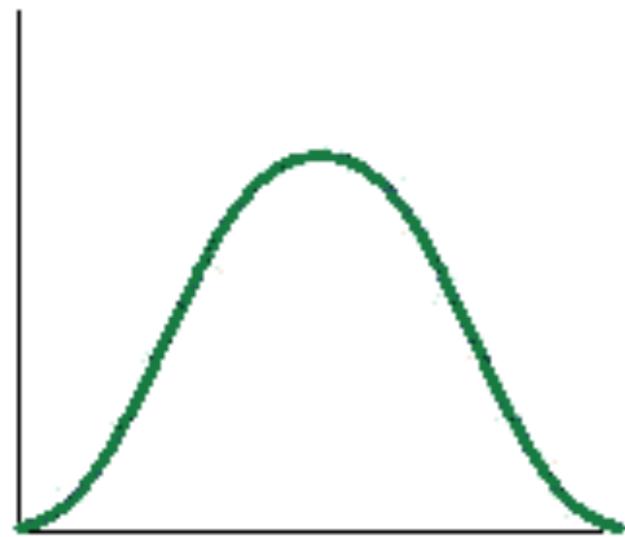
Notepad



pico



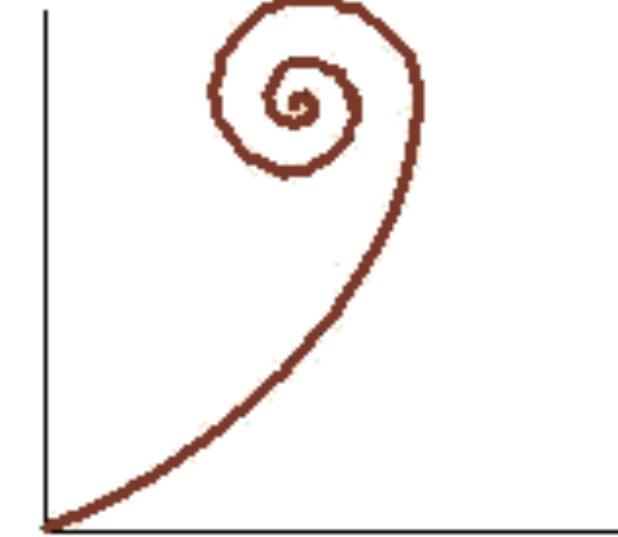
Visual Studio



vi



emacs



vi



and yet ...

"i love vim" – Google Search

http://www.google.com/search?hl=en&safe=off&gbv=2&bv...  
1P 26R DDG G Gm Gr Gt Gd Gv Subscribe... AWS Activity Twitter Search  
"i love vim" – Google Search

Web Images Videos Maps News Shopping Gmail more ▾ BillOdom@gmail.co

Google "i love vim" Search

About 116,000 results (0.15 seconds) Advanced search

Everything [zerosum dirt\(nap\) - Why I Love Vim](#) ☆

# The Sales Pitch

The Sales Pitch✓

# Philosophy

*Use a single editor well.*

*The editor should be an extension of your hand;  
make sure your editor is configurable,  
extensible, and programmable.*

—*The Pragmatic Programmer*  
by Andrew Hunt and David Thomas

Save Time

# Increase Productivity

# Reduce Tedium

Remain Sane

# Goals

# Getting Started

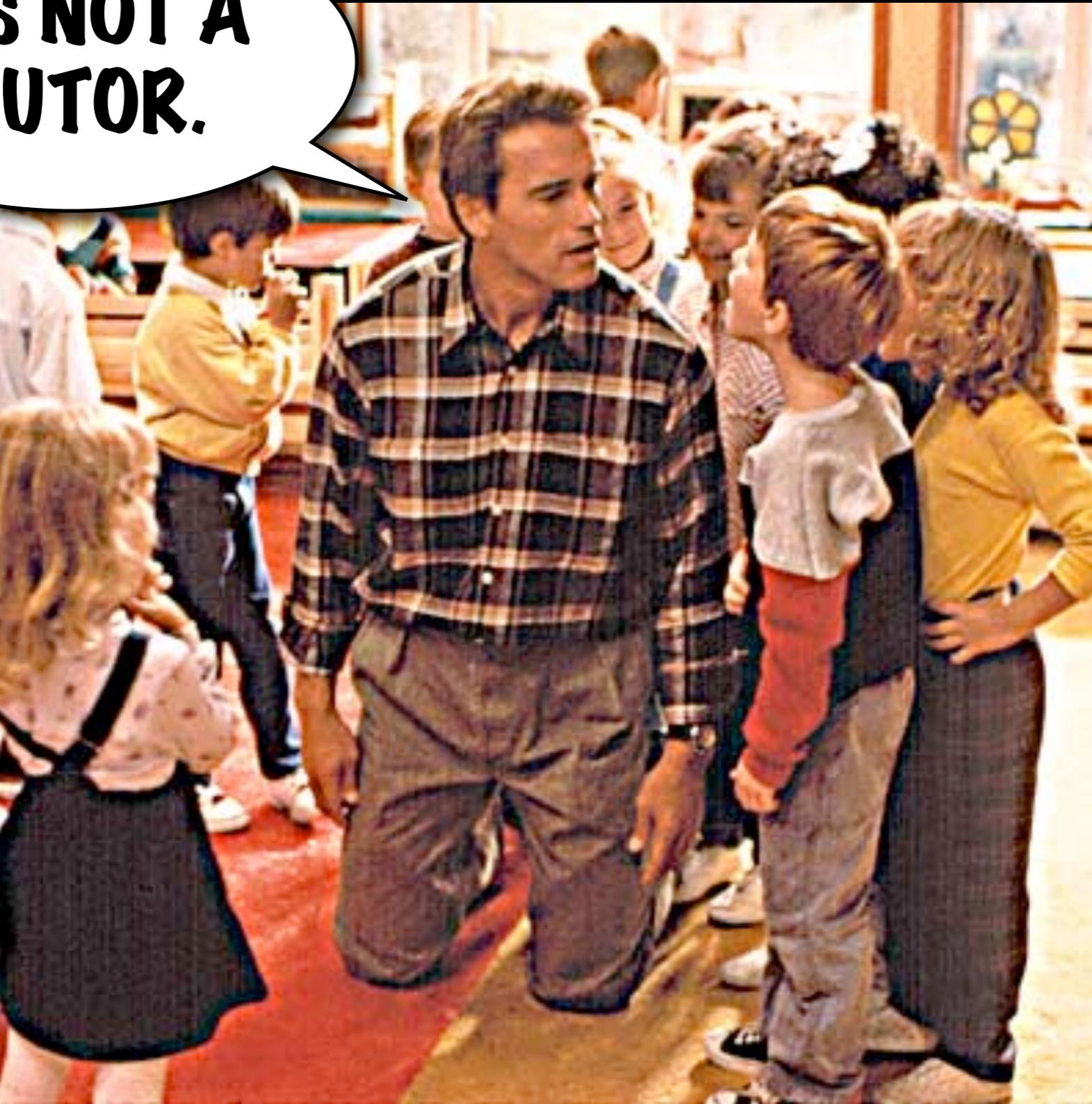
# A Few Big Ideas

# Lots of Tips

Discussion  
Questions  
Comments

# Non-Goals

**IT'S NOT A  
TUTOR.**



Editor war – Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Editor\_war

Reader

editor wars

Editor war – Wikipedia, the free e...

New features Log in / create account

Article Discussion Read Edit View history Search

**Editor war**

From Wikipedia, the free encyclopedia

*For a type of conflict between wiki editors, see [Edit war](#).*

**Editor war** is the common name for the rivalry between users of the [vi](#) and [Emacs](#) text editors. The rivalry has become a lasting part of [hacker culture](#) and the [free software community](#).

Many [flame wars](#) have been fought between groups insisting that their editor of choice is the paragon of editing perfection, and insulting the others. Unlike the related battles over [operating systems](#), [programming languages](#), and even source code [indent style](#), choice of editor usually only affects oneself.

**Contents** [hide]

- 1 Differences between vi and Emacs
  - 1.1 Benefits of vi-like editors
  - 1.2 Benefits of Emacs
- 2 Humour
- 3 Current state of the editor war
- 4 References
- 5 External links

“Escape Meta Alt Control Shift”

“Emacs: A great  
operating system,  
lacking only a decent  
editor.”

vi has two modes:  
"beep repeatedly" and  
"break everything"

*Sometimes, people ask me if it is a sin in  
the church of Emacs to use the editor Vi. It  
is true that Vi-Vi-Vi is the editor of the  
beast. But using a free version of Vi is not a  
sin but a penance.*

— Richard Stallman (creator of EMACS)



# Getting Started

Step One: Don't.

*A small piece of advice: Don't try to do it all at once.*

*I attempted to learn everything immediately, to instantly become the magician that other vi users seemed to be. I'd become overwhelmed and frustrated, and then give up.*

*Don't do that. Get a good version of vi that doesn't make you abandon everything you already know, then slowly, steadily build up your repertoire of magic tricks. I've been using some form of vi for years, and I still learn new stuff about it all the time. Frankly, that's part of its appeal – it's as deep as I want to go, and there's more power every time I take a few minutes to learn it.*

– an email from Bill Odom to Merlin Mann

# Getting Vim

UNIX or Linux?  
Done.

Mac OS X:  
[code.google.com/p/macvim/](http://code.google.com/p/macvim/)

macvim – Project Hosting on Google Code

Reader [macvim](#)

macvim – Project Hosting on Goo...

BillOdom@gmail.com | My favorites ▾ | Profile | Sign out

**macvim**  
Vim for the Mac

Project Home Downloads Wiki Issues Source

Summary | [Updates](#) | [People](#)

MacVim is a port of the text editor Vim to Mac OS X.

Starred ([view starred projects](#))

Activity: High

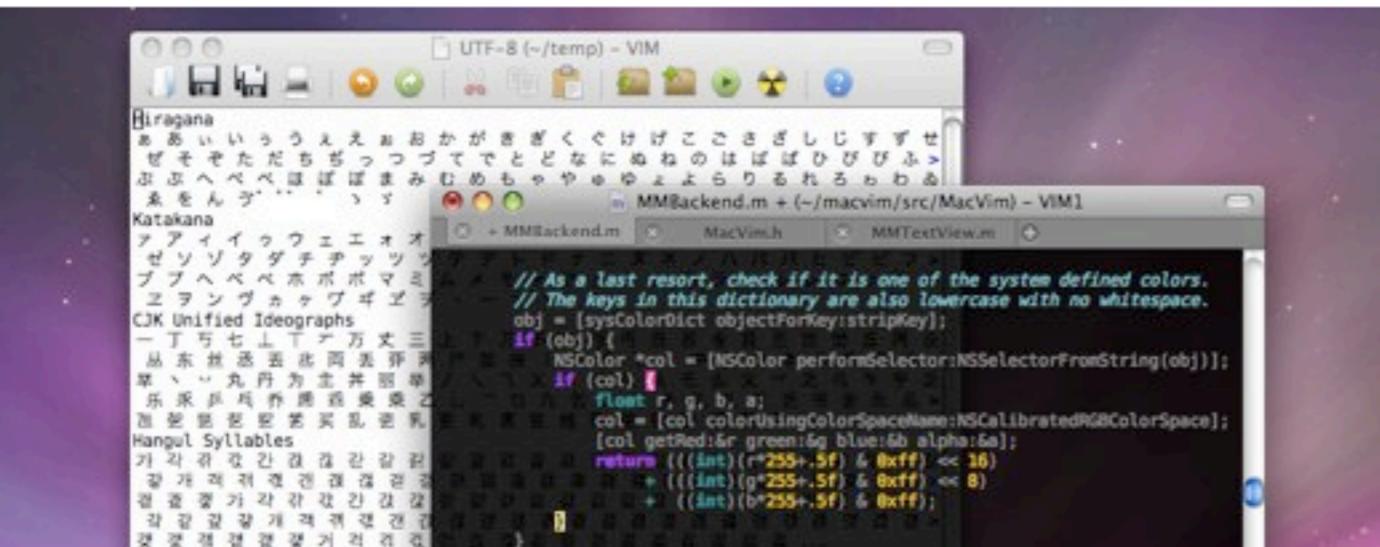
Code license: [GNU General Public License v2](#)

Labels: vim, editor, mac, osx, cocoa, objective-c

Featured wiki pages:  
[ReportingBugs](#)  
[Show all »](#)

Blogs:  
[Björn's Blog](#)

External links:  
[Source code repository](#)



MacVim supports multiple windows with tabbed editing and a host of other features such as:

- bindings to standard OS X keyboard shortcuts (⌘Z, ⌘V, ⌘A, ⌘G, etc.),
- transparent backgrounds,

Everybody else:  
[www.vim.org/download.php](http://www.vim.org/download.php)

# Important Settings

```
:set nocompatible
```

:set incsearch

:set hlsearch

Categorize.pm (~/Work/List-Categorize/src/lib>List) – VIM1

```
1 package List::Categorize;
2 #
3 # ABSTRACT: Categorize list items into a hash of named sublists.
4 #
5 # See documentation after __END__ below.
6 #
7
8 use strict;
9 use warnings;
10
11 ## Module Interface
12
13 use base 'Exporter';
14 our $VERSION      = '0.04';
15 our @EXPORT_OK    = qw( categorize );
16 our %EXPORT_TAGS = ( all => \@EXPORT_OK );
17
18
19 ## Subroutines
20
21 sub categorize (&@)
22 #
src/lib>List/Categorize.pm          1,15          Top
/cate
```

:set ignorecase

:set smartcase

Categorize.pm (~/Work/List-Categorize/src/lib>List) – VIM1

```
1 package List::Categorize;
2 #
3 # ABSTRACT: Categorize list items into a hash of named sublists.
4 #
5 # See documentation after __END__ below.
6 #
7
8 use strict;
9 use warnings;
10
11 ## Module Interface
12
13 use base 'Exporter';
14 our $VERSION      = '0.04';
15 our @EXPORT_OK    = qw( categorize );
16 our %EXPORT_TAGS = ( all => \@EXPORT_OK );
17
18
19 ## Subroutines
20
21 sub categorize (&@)
22 #
src/lib>List/Categorize.pm          1,15          Top
/Cate
```

```
:set tabstop=4
```

```
:set softtabstop=4
```

:set expandtab

:set smarttab

```
:set shiftwidth  
:set shiftround
```

:set vb

:set t\_vb=

:set hidden

```
:set backspace=  
indent,eol,start
```

```
:set whichwrap+=<,>,[,]
```

```
:set backupdir=.
:set backupdir^=~/tmp, /tmp
:set directory=~/tmp//, .
```

```
:set backupdir=.
:set backupdir^=$TEMP
:set directory=$TEMP\\\\\\
```

```
:set selection=exclusive  
:set selectmode=mouse,key  
:set mousemodel=popup  
:set keymodel=  
    startsel,stopsel
```

:behave mswin

:filetype plugin indent on

# **Life-Changing Keystrokes**



The screenshot shows a VIM1 window with the file `Categorize.pm` open. The code is a Perl module named `List::Categorize`. It includes documentation for categorizing lists into hash sublists, strict and warnings pragmas, base class Exporter, and subroutine categorize. The code is color-coded for readability.

```
1 package List::Categorize;
2 #
3 # ABSTRACT: Categorize list items into a hash of named sublists.
4 #
5 # See documentation after __END__ below.
6 #
7
8 use strict;
9 use warnings;
10
11 ## Module Interface
12
13 use base 'Exporter';
14 our $VERSION      = '0.04';
15 our @EXPORT_OK    = qw( categorize );
16 our %EXPORT_TAGS = ( all => \@EXPORT_OK );
17
18
19 ## Subroutines
20
21 sub categorize (&@)
22 #
src/lib/List/Categorize.pm          8,1           Top
/\<use\>
```

Ctrl N

Ctrl P

Categorize.pm + (~/Work/List-Categorize/src/lib>List) - VIM1

```
|| 58     local $_ = $element;
|| 59             category
|| 60     # Execute th Categorize
|| 61     # for this e categorize
|| 62     #
|| 63     my $category categorizer
|| 64             categorizing
|| 65     # If a categ Categories /System/Library/Perl/5.10.0/warnings.pm
|| 66     # hash, and catmask /System/Library/Perl/5.10.0/warnings.pm
|| 67     # that key. catfile /System/Library/Perl/5.10.0/File/Spec.p
|| 68     # catdir /System/Library/Perl/5.10.0/File/Spec/F
|| 69     # If the cat catpath /System/Library/Perl/5.10.0/File/Spec/F
|| 70     # then leave catch /System/Library/Perl/5.10.0/darwin-thre
|| 71     # categories cat_decode /System/Library/Perl/5.10.0/darwin-thre
|| 72     if (defined $category) categories /System/Library/Perl/5.10.0/Exporter.pm
2- 72
23 73 {
23 74     push @{$$sublists{ $category } }, @_;
23 75 }
|| 76 }
|| 77
|| 78     return %sublists;
|| 79 }
```

src/lib>List/Categorize.pm [+] 72,25 29%

-- Keyword completion (^N^P) match 1 of 13



Categorize.pm (~/Work/List-Categorize/src/lib>List) – VIM1

Categorize.pm

```
|| 77
|| 78      return %sublists;
|| 79 }
|| 80
|| 81
```

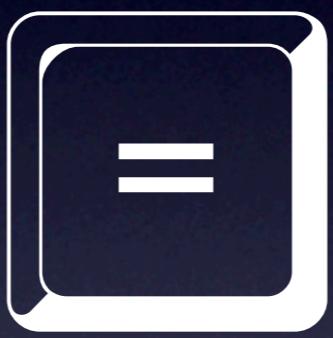
src/lib>List/Categorize.pm 79,1 36%

Completion can be done for:

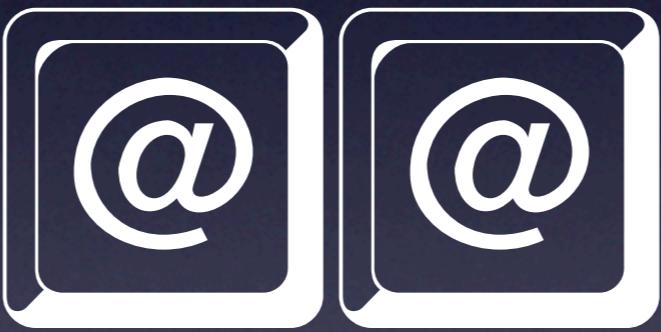
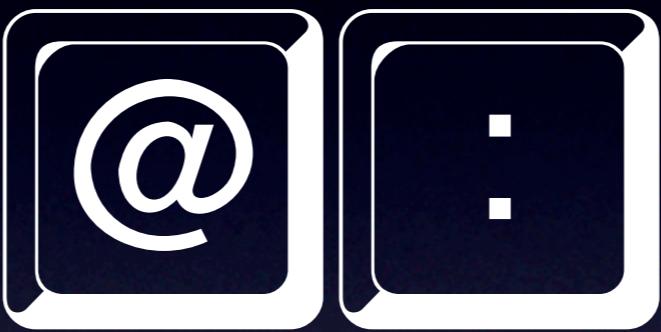
1. Whole lines	i_CTRL-X_CTRL-L
2. keywords in the current file	i_CTRL-X_CTRL-N
3. keywords in 'dictionary'	i_CTRL-X_CTRL-K
4. keywords in 'thesaurus', thesaurus-style	i_CTRL-X_CTRL-T
5. keywords in the current and included files	i_CTRL-X_CTRL-I
6. tags	i_CTRL-X_CTRL-]
7. file names	i_CTRL-X_CTRL-F
8. definitions or macros	i_CTRL-X_CTRL-D
9. Vim command-line	i_CTRL-X_CTRL-V
10. User defined completion	i_CTRL-X_CTRL-U
11. omni completion	i_CTRL-X_CTRL-O
12. Spelling suggestions	i_CTRL-X_s
13. keywords in 'complete'	i_CTRL-N

insert.txt [Help][R0] 576,18-40 30%









g

G

g j

g k

# Custom Key Mappings

`$VIMRUNTIME/mswin.vim`

```
:vmap <Tab>      >gv  
:vmap <S-Tab>    <gv
```

```
:nnoremap <F7>      gqap  
:nnoremap <S-F7>    vipJ
```

[No Name] + - VIM1

[No Name] [+]

```
1
2 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed eu lorem s>
3
4 Fusce placerat dapibus purus. Proin id pede eu justo malesuada vehicula.>
5
6 Quisque dapibus turpis ut elit. In sem. Pellentesque eget arcu nec ipsum>
7
8 Quisque tincidunt. In hac habitasse platea dictumst. Aenean porta congue>
9
10 Morbi congue. Sed at enim. Morbi ac turpis sit amet nulla pretium ornare>
11
12 Etiam sodales. Morbi ac lacus. Aenean facilisis eleifend erat. Ut ac ero>
13
14 Nulla adipiscing justo in ligula. Maecenas sollicitudin ligula ultricies>
15
16 Aliquam nulla felis, lobortis vitae, iaculis eget, molestie eu, nibh. Cu>
17
18 Cras sapien lorem, posuere volutpat, posuere ut, aliquet eget, elit. Cla>
19
20 Donec a dolor quis ipsum cursus interdum. In tincidunt vehicula lectus. >
21
```

~

[No Name] [+] 2,1 All

"~/Work/vim-sandbox/samples/sample-text-unwrapped.txt" 20L, 6714C

[No Name] + - VIM1

[No Name] [+]

```
1 
2 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed eu
3 lorem suscipit ipsum accumsan molestie. Phasellus iaculis enim
4 vel sem. Maecenas vel nulla in justo fringilla mollis. Cras
5 congue ullamcorper quam. Morbi ac massa. Vestibulum interdum eros
6 nec sem auctor molestie. Praesent urna sem, lobortis et, rutrum
7 feugiat, laoreet at, odio. Etiam elementum orci sed leo.
8 Suspendisse ac elit. Phasellus lorem turpis, tempus at, aliquet
9 sed, semper nec, justo. Cras et enim sit amet mauris tincidunt
10 luctus. Aliquam vulputate. Curabitur fringilla. In quis purus.
11 
12 Fusce placerat dapibus purus. Proin id pede eu justo malesuada vehicula.>
13 
14 Quisque dapibus turpis ut elit. In sem. Pellentesque eget arcu nec ipsum>
15 
16 Quisque tincidunt. In hac habitasse platea dictumst. Aenean porta congue>
17 
18 Morbi congue. Sed at enim. Morbi ac turpis sit amet nulla pretium ornare>
19 
20 Etiam sodales. Morbi ac lacus. Aenean facilisis eleifend erat. Ut ac ero>
21 
22 Nulla adipiscing justo in ligula. Maecenas sollicitudin ligula ultricies>
```

[No Name] [+]

1,0-1 Top

```
:nnoremap n      nzz
:nnoremap N      Nzz
:nnoremap *      *zz
:nnoremap #      #zz
:nnoremap g*    g*zz
:nnoremap g#    g#zz
```

```
inoremap <C-U>=
<Esc>kyyyp^v$r=ja
```

A screenshot of a VIM editor window titled "[No Name] + - VIM1". The window contains the following text:

```
1
2 Header
3
4 Subheader 1
5
6 Plain text.
7
8
```

The word "Header" is highlighted in green. The status bar at the bottom shows the current file name "[No Name]" and line number "2,1". The search bar on the right displays the text "All".

The screenshot shows a VIM editor window with the title bar "[No Name] + - VIM1". The main area displays the following text:

```
1
2 Header
3 =====
4
5 Subheader 1
6 -----
7
8 Plain text.
9
10
```

The text is numbered from 1 to 10. Lines 1 through 10 are visible in the main editor area, while lines 11 through 20 are shown as a series of tilde (~) characters. A vertical scroll bar is located on the right side of the editor area.

The status bar at the bottom of the window contains the following information:

- [No Name] [+]
- 1,0-1
- All

A horizontal scroll bar is located at the bottom of the status bar area.

# Important Features

:help

:helpgrep

# Text Objects

# Folding



# Filtering

[No Name] + - VIM1

[No Name] [+]

categorize.t

[No Name] [+]

```
1 {
2 "glossary": {
3 "title": "example glossary",
4 "GlossDiv": {
5 "title": "S",
6 "GlossList": {
7 "GlossEntry": {
8 "ID": "SGML",
9 "SortAs": "SGML",
10 "GlossTerm": "Standard Generalized Markup Language",
11 "Acronym": "SGML",
12 "Abbrev": "ISO 8879:1986",
13 "GlossDef": {
14 "para": "A meta-markup language, used to create markup languages such as",
15 "GlossSeeAlso": ["GML", "XML"]
16 },
17 "GlossSee": "markup"
18 }
19 }
20 }
21 }
22 }
```

[No Name] [+]

1,1

Top

[No Name] + - VIM1

[No Name] [+]

categorize.t

[No Name] [+]

```
1 {
2     "glossary" : {
3         "GlossDiv" : {
4             "GlossList" : {
5                 "GlossEntry" : {
6                     "ID" : "SGML",
7                     "GlossTerm" : "Standard Generalized Markup Language",
8                     "GlossDef" : {
9                         "para" : "A meta-markup language, used to create markup languages like XML and GML.",
10                        "GlossSeeAlso" : [
11                            "GML",
12                            "XML"
13                        ]
14                    },
15                    "SortAs" : "SGML",
16                    "Acronym" : "SGML",
17                    "Abbrev" : "ISO 8879:1986",
18                    "GlossSee" : "markup"
19                }
20            },
21            "title" : "S"
22        },
23    }
24}
```

[No Name] [+]

1,1

Top

:%!/bin/format-json.pl

# Registers

# Macros

# Visual Blocks

spinconfig.xml (~/Work/FlexSpin) - VIM1

```
78 <priority="240" />
79
80
81 <priority="110" />
82 <priority="70" />
83 <priority="110" />
84 <priority="5" />
85 <priority="15" />
86 <priority="12" />
87 <priority="8" />
88 <priority="20" />
89 <priority="40" />
90 <priority="140" />
91 <priority="140" />
92 <priority="140" />
93 <priority="140" />
94 <priority="140" />
95 <priority="140" />
96 <priority="130" />
97 <priority="130" />
98 <array="true">
99 <array="true">
```

~/Work/FlexSpin/spinconfig.xml 94,184 59%

-- VISUAL BLOCK -- 14x16

# Useful Plugins

# NERD Tree

" Press ? for help

.. (up a dir)  
/Users/bodom/Work/List-Cat>

|+releases/  
|+scratch/  
|~src/  
| |+lib/  
| |+List-Categorize-0.04/  
| |~t/  
| | | -00-load.t  
| | | -categorize.t  
| | | -critique.t  
| | | -perlriticrc  
| | | -pod-coverage.t  
| | | '-pod.t  
| | -Changes  
| | -dist.ini  
| | -List-Categorize-0.04.t>  
| | '-README  
| '-todo.txt  
~

<bodom/Work>List-Categorize src/t/categorize.t 19,1 Top

```
1 #! perl -T
2 #
3 # categorize.t
4 #
5 # Tests for List::Categorize::categorize().
6 #
7
8 use strict;
9 use warnings;
10
11 use Test::More tests => 14;
12 use Test::Deep;
13 use Test::NoWarnings;
14
15 use List::Categorize qw( categorize );
16
17
18 # List of products, IDs, and prices. Used as
19 #
20 # Note: The order of this list, and of the "
21 # below, is important. The comparison function
22 # ordering of the sublists.
```

Command-T

GoToFile - (~/Work/List-Categorize) - VIM1

GoToFile [-] +

```
1 #! perl -T
2 #
src/t/categorize.t          1,1      Top
> src/lib>List/Categorize.pm
src/List-Categorize-0.04.tar.gz
src/List-Categorize-0.04/README
src/List-Categorize-0.04/Changes
src/List-Categorize-0.04/LICENSE
src/List-Categorize-0.04/t/pod.t
releases/List-Categorize-0.03.tar
src/List-Categorize-0.04/MANIFEST
src/List-Categorize-0.04/META.yml
src/List-Categorize-0.04/dist.ini
releases/List-Categorize-0.01.tar.gz
releases/List-Categorize-0.02.tar.gz
src/List-Categorize-0.04/Makefile.PL
src/List-Categorize-0.04/t/00-load.t
src/List-Categorize-0.04/t/critique.t
src/List-Categorize-0.04/t/categorize.t
src/List-Categorize-0.04/t/perlriticrc
src/List-Categorize-0.04/t/pod-coverage.t
src/List-Categorize-0.04/lib>List/Categorize.pm
```

GoToFile 1,1 All
>> 1

GoToFile - (~/Work/List-Categorize) - VIM1

GoToFile [-] +

```
1 #! perl -T
2 #
src/t/categorize.t          1,1      Top
> src/lib>List/Categorize.pm
src/List-Categorize-0.04/Changes
src/List-Categorize-0.04/t/critique.t
src/List-Categorize-0.04/t/categorize.t
src/List-Categorize-0.04/lib>List/Categorize.pm
src/List-Categorize-0.04.tar.gz
src/List-Categorize-0.04/README
src/List-Categorize-0.04/LICENSE
src/List-Categorize-0.04/t/pod.t
releases/List-Categorize-0.03.tar
src/List-Categorize-0.04/MANIFEST
src/List-Categorize-0.04/META.yml
src/List-Categorize-0.04/dist.ini
releases/List-Categorize-0.01.tar.gz
releases/List-Categorize-0.02.tar.gz
src/List-Categorize-0.04/Makefile.PL
src/List-Categorize-0.04/t/00-load.t
src/List-Categorize-0.04/t/perlcriticrc
src/List-Categorize-0.04/t/pod-coverage.t
```

GoToFile 1,1 All
>> lc

GoToFile - (~/Work/List-Categorize) - VIM1

```
1 #! perl -T
2 #
3 # categorize.t
4 #
5 # Tests for List::Categorize::categorize().
6 #
7
8 use strict;
9 use warnings;
10
11 use Test::More tests => 14;
12 use Test::Deep;
13 use Test::NoWarnings;
14
15 use List::Categorize qw( categorize );
```

src/t/categorize.t 1,1 Top

> src/List-Categorize-0.04/t/pod.t  
src/List-Categorize-0.04/t/perlcriticrc  
src/List-Categorize-0.04/t/pod-coverage.t  
src/lib>List>Categorize.pm  
src/List-Categorize-0.04/lib>List>Categorize.pm  
src/List-Categorize-0.04/Makefile.PL

GoToFile 1,1 All

>> lcp

GoToFile - (~/Work/List-Categorize) - VIM1

```
1 #! perl -T
2 #
3 # categorize.t
4 #
5 # Tests for List::Categorize::categorize().
6 #
7
8 use strict;
9 use warnings;
10
11 use Test::More tests => 14;
12 use Test::Deep;
13 use Test::NoWarnings;
14
15 use List::Categorize qw( categorize );
16
17
18 # List of products, IDs, and prices. Used as sample data for several tes>
19 #
```

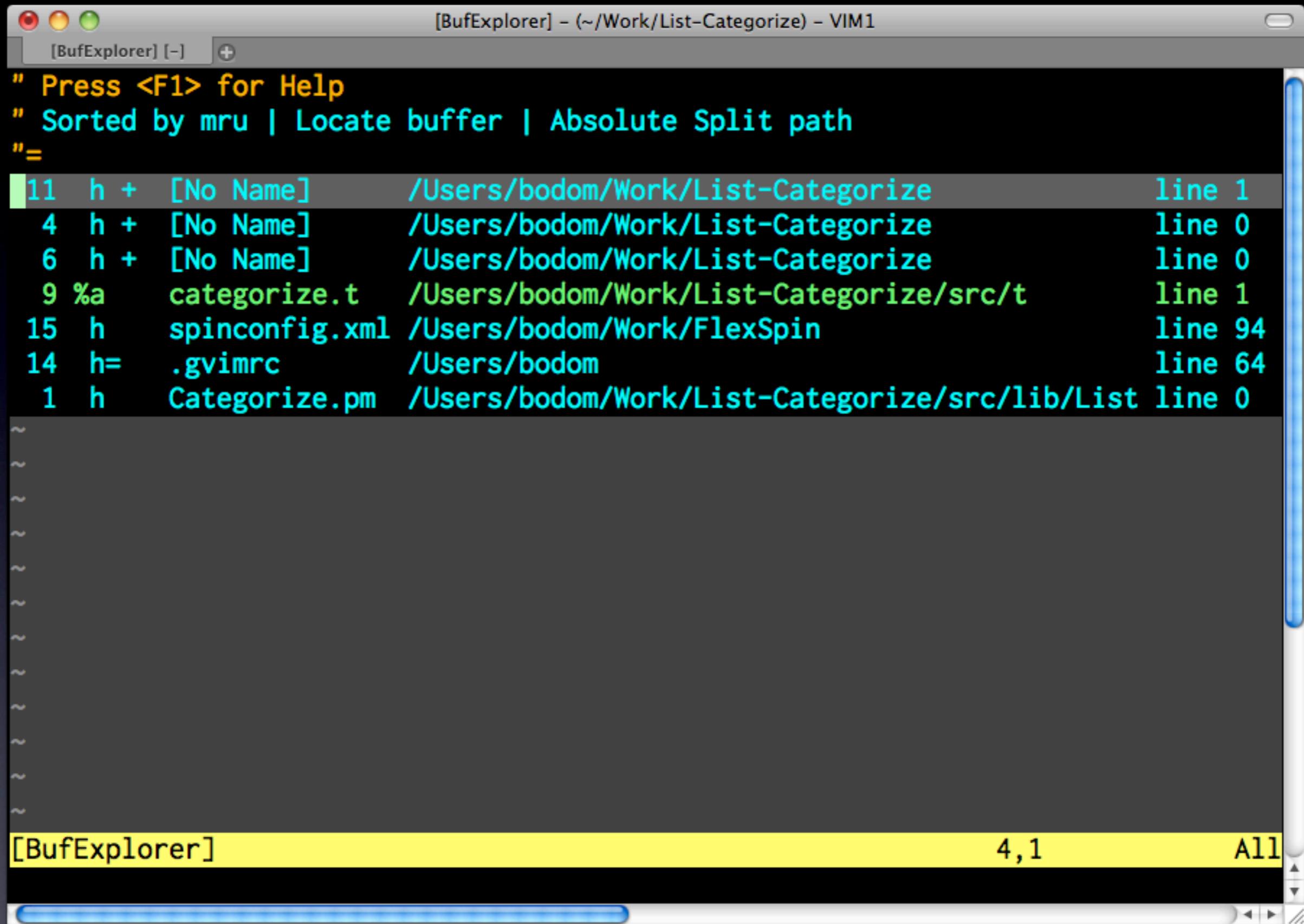
src/t/categorize.t 1,1 Top

> src/lib>List/Categorize.pm src/List-Categorize-0.04/lib>List/Categorize.pm

GoToFile 1,1 All

>> lcpm

# Bufexplorer



# Taglist

\_\_Tag\_List\_\_ - (~/Work/List-Categorize) - VIM1

```
1 #! perl -T
2 #
3 # categorize.t
4 #
5 # Tests for List::Categorize::categorize(>
6 #
7
8 use strict;
9 use warnings;
10
11 use Test::More tests => 14;
12 use Test::Deep;
13 use Test::NoWarnings;
14
15 use List::Categorize qw( categorize );
16
17
18 # List of products, IDs, and prices. Used>
19 #
20 # Note: The order of this list, and of th>
21 # below, is important. The comparison fun>
22 # ordering of the sublists.
```

" Press <F1> to display he>

- Categorize.pm (/Users/bodo>

|- package

|- List::Categorize

|- subroutine

|- categorize

- categorize.t (/Users/bodom>

|- subroutine

|- test\_list\_of\_simple\_sc>

|- test\_list\_of\_hashrefs\_>

|- test\_list\_of\_hashrefs\_>

|- test\_simple\_size\_limit>

|- test\_list\_of\_hashrefs\_>

|- test\_list\_of\_lists

|- test\_empty\_list

|- test\_sublist\_element\_t>

|- test\_collapsing\_sublis>

|- test\_ignored\_elements

|- test\_categorizer\_args

src/t/categorize.t 1,1 Top \_\_Tag\_List\_\_ 10,1 Top

# Customizations

# Extra Filetypes

# Syntax-Sensitive Search

# Literal Search

# Smart Spacing

# Quote Switcher

Paste Escaped Text

# Message Redirector

“X-Ray Mode”

categorize.t (~/Work/List-Categorize/src/t) - VIM1

```
1 #! perl -T
2 #
3 # categorize.t
4 #
5 # Tests for List::Categorize::categorize().
6 #
7
8 use strict;
9 use warnings;
10
11 use Test::More tests => 14;
12 use Test::Deep;
13 use Test::NoWarnings;
14
15 use List::Categorize qw( categorize );
16
17
18 # List of products, IDs, and prices. Used as sample data for several tes>
19 #
20 # Note: The order of this list, and of the "expected" lists in the tests
21 # below, is important. The comparison functions check for proper
22 # ordering of the sublists.
```

src/t/categorize.t 11,26 Top :q

categorize.t (~/Work/List-Categorize/src/t) - VIM1

```
1 #! perl -T$  
2 #$  
3 # categorize.t$  
4 #$  
5 # Tests for List::Categorize::categorize().$  
6 #$  
7 $  
8 use strict;$  
9 use warnings;$  
10 $  
11 use Test::More tests => 14;$  
12 use Test::Deep;$  
13 use Test::NoWarnings;$  
14 $  
15 use List::Categorize qw( categorize );$  
16 $  
17 $  
18 # List of products, IDs, and prices. Used as sample data for several tes>  
19 #$  
20 # Note: The order of this list, and of the "expected" lists in the tests$  
21 # below, is important. The comparison functions check for proper$  
22 # ordering of the sublists.$
```

src/t/categorize.t 11,26 Top  
/Test

# Integration

```
:setlocal makeprg=perl -c
```

: compiler perl

: compiler perlcritic

```
:set equalprg=perltidy  
:set keywordprg=perldoc
```

# The Future

# Snipmate

fugitive.vim

“Stealing from Padre”

# Resources

- **Vim**  
<http://www.vim.org/>
- **MacVim**  
<http://code.google.com/p/macvim/>
- **Vim Plugins, Add-Ons, Scripts**  
<http://www.vim.org/scripts/index.php>
- **Vim Tips Wiki**  
[http://vim.wikia.com/wiki/Vim\\_Tips\\_Wiki](http://vim.wikia.com/wiki/Vim_Tips_Wiki)
- **Efficient Editing With Vim**  
<http://jmcpherson.org/editing.html>

- **Vimcasts**  
<http://vimcasts.org/>
- **Derek Wyatt's Vim Tutorial Videos**  
<http://www.derekwyatt.org/vim/vim-tutorial-videos/>
- **Zenclavier: Extreme Keyboarding**  
[http://www.oreilly.com/news/zenclavier\\_1299.html](http://www.oreilly.com/news/zenclavier_1299.html)
- **Your problem with Vim is that you don't grok vi**  
<http://stackoverflow.com/questions/1218390/#1220118>
- **Damian Conway's “Scripting the Vim Editor” Series**  
<https://www.ibm.com/developerworks/linux/library/l-vim-script-l/>

Thanks!