Expert Code Grokking with Vim

Vim hacks for penetration testers

David Thiel

November 21, 2017

Outline

Hello.

Vim is a useful tool.

It excels at source review.

It takes a while to learn.

Hopefully, this will make it faster.

Prerequisites

- Vim
- Basic vi knowledge
  - If you don't have this, try: `http://www.openvim.com/tutorial.html`
- Git, for fetching plugins
- A basic `vimrc` and a `.vim` directory

The Basics

```
                              basics
syntax on               " enable syntax highlighting
filetype plugin on      " filetype detection
filetype indent on
set number              " line numbers
set hidden              " allow invisible buffers
set ignorecase          " case-insensitive searching
set smartcase           " but be smart about it
set hlsearch            " highlight all search matches
set incsearch           " search incrementally
set et                  " expand tabs to spaces
set tags=./tags         " ctags - we'll get into that
```

Basic Vim Concepts

Modes

- Two main modes: *command* (aka "normal") and *insert*
  - Command mode is for giving instructions to vim.
- Other modes:
  - ex mode: Entered by typing "`:`", this is vim's "command line"
  - Visual: Visually select blocks of text, by using `Shift-V` (line mode) `CTRL-v`, (block mode), or "`v`" plus operators[1][2]

[1] http://vimdoc.sourceforge.net/htmldoc/visual.html#visual-operators

[2] http://stackoverflow.com/a/1218429

Vim's hierarchy of buffers

- A "buffer" holds the contents of files.
- A "window" is a portal to a buffer.
- A "tab" is a container of windows.
  - You may think you want to only use tabs, but you don't.
  - Avoid them until you master buffers and windows.

Vim’s hierarchy of buffers

- A “buffer” holds the contents of files.
- A “window” is a portal to a buffer.
- A “tab” is a container of windows.
  - You may think you want to only use tabs, but you don’t.
  - Avoid them until you master buffers and windows.

Vim’s hierarchy of buffers

- A “buffer” holds the contents of files.
- A “window” is a portal to a buffer.
- A “tab” is a container of windows.
  - You may think you want to only use tabs, but you don’t.
  - Avoid them until you master buffers and windows.

Vim’s hierarchy of buffers

- A “buffer” holds the contents of files.
- A “window” is a portal to a buffer.
- A “tab” is a container of windows.
  - You may think you want to only use tabs, but you don’t.
  - Avoid them until you master buffers and windows.

Vim’s hierarchy of buffers

- A “buffer” holds the contents of files.
- A “window” is a portal to a buffer.
- A “tab” is a container of windows.
  - You may think you want to only use tabs, but you don’t.
  - Avoid them until you master buffers and windows.

Basic Vim Concepts

Buffers

- Use `:ls` to show buffers. Many people also use a buffer manager such as BufTabs.
- Load a new buffer without viewing it with `:badd`.
- Note: The `:quit` command means to close a window. `:bdelete` / `:bd` means to delete/close a buffer.

Basic Vim Concepts

Windows

- Opening/closing windows:
  - `CTRL-w s` or `:split (filename)` — new horizontal split
  - `CTRL-w v` or `:vsplit (filename)` — new vertical split
  - `CTRL-w o` or `:only` — close all other windows
  - `CTRL-w c` — close window
- Navigating/moving windows:
  - `CTRL-w h/j/k/l` — change to window in that direction
  - `CTRL-w H/J/K/L` — move current window to (direction) side of the screen
  - `CTRL-w r` — rotate windows

Basic Vim Concepts

Jumps

- When you “jump” to another part of a file, your old position is stored in the jumplist
- Things that make jumps:
  - Jumping to a search result
  - Changing to a new buffer
  - Jumping to symbol definition
- Navigate your jumplist with `(CTRL-o)` and `(CTRL-i)`

Basic Vim Concepts

Changes

- Similar to jumps, but for lines that were changed
- See change list with `:changes`
- `g;` goes to the position of the last change
- `g,` goes back up the change list

Basic Vim Concepts

Named Registers

- Named registers can be used for storing lots of things
- IMO, the most useful is using registers for “complex repeats”, kind of an insta-macro
- Usually, Last change is repeated with “`.`”
- Complex repeats allow repeating very complex command sequences
- Usage:
  - `qa` starts recording into register “a”
  - Perform whatever complex sequence of commands and movements you feel like
  - When finished, hit `q` again.
  - To execute the contents of this register, call `@a`
  - `@@` to repeat the last executed register

Basic Vim Concepts

Other Registers

- There are also "numbered registers"
  - Used for remembering yanks/deletes
  - By default, register 1 has your most recent yank/delete
  - Access yank-before-last by `"2p`, and so on
- The `/` register: holds last search pattern
- The `_` register: blackhole — delete things to this register with `"_d` to have them not affect your delete/yank history
- There's a lot more you can do with registers; check http://blog.sanctum.geek.nz/advanced-vim-registers/

Basic Vim Concepts

Undos

- Vim carries undo actions in a tree.
- You can make a change, undo it, make another change. You have now branched.
- This is hard to understand, but take my word for it.
- `u` and `(CTRL-r)` undo and redo along the *main branch*.
- `g+` and `g-` move *forward and backward in time*.
- Gundo can help you visualize this. See the screencast.

Basic Vim Concepts

Marks

- “marks” are pointers to specific locations in specific files.
- Simple usage:
  - `ma` to make mark “a”
  - `'a` to jump to the line mark “a” is on
  - `` `a `` to jump to the exact position of mark “a”
  - `:delm a` to delete mark “a”
  - Use “A” instead of “a” and this will make a cross-file mark — you can jump to it at any time, regardless of whether you’re editing that file at the moment
  - You can use motions with marks: `d'a` deletes from your current position to mark “a”
- But wait! There are other, cooler things.

Basic Vim Concepts

Marks

- Try using the `:marks` command.
- Note there are some special marks:
  - `.` — location of last change
  - `'` — the place you were before your last jump
  - `0` — the location and file you were at when you last quit vim (it's a stack: you can also use 1-9)
- Check out `:help mark-motions` for more

Sessions and Views

Views

- Vim has a concept of “views”, which specify where you last were in a file
- You can configure a filetype to do this thusly:

Saving and loading Python views

```
augroup python
au BufEnter *.py,*.pyw set smartindent smarttab nospell
au BufWinLeave *.py mkview
au BufWinEnter *.py silent loadview
augroup end
```

- Note that you can screw yourself up with this; if you notice some change in your vimrc isn’t taking effect, try nuking the file in your viewdir.

Sessions and Views

Sessions

- This does save potentially sensitive data (filenames)
- I recommend storing it outside of your .vim directory:

Change viewdir

```
set viewdir=$HOME/.views
```

- You can also save your whole vim session state with `:mksession`
- This writes your state out to `Session.vim` in the cwd
- Includes all your open files, panels, etc
- Restore with `:so Session.vim`
- Smoother session integration can be had with https://github.com/tpope/vim-obsession

Interlude

Putting things together

- visually select a word: `vaw`
- visually select a sentence: `vas`
- visually select a paragraph: `vap`
- visually select a block of C code: `va{`
- visually select from here to a search string: `v/someword`

- Now, think what happens if you use “`y`”, “`c`” or “`d`” here…
- Before *or* after: `vapy` == `yap`
- You see how your life is changing now

Note: using “`i`” (meaning “inner”) instead of “`a`” is probably more common in practice. Try both.

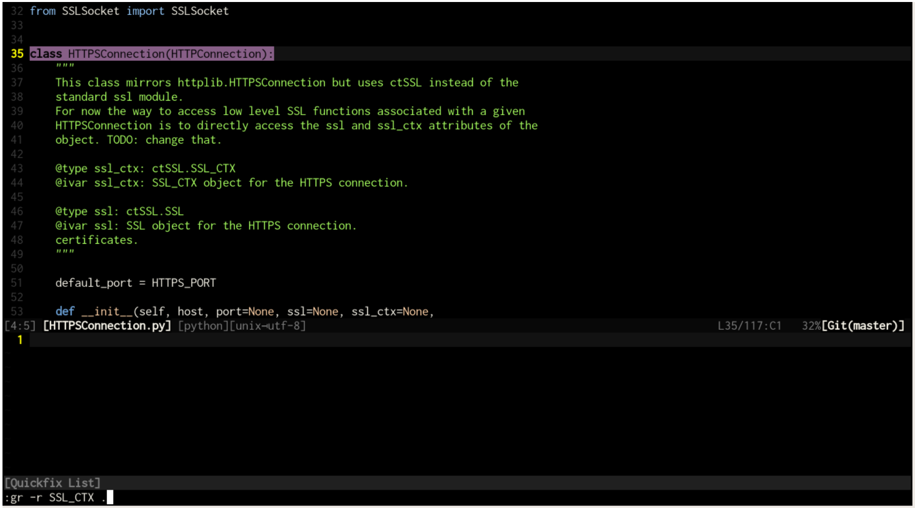
The Quickfix window

Your best friend

- Quickfix takes lists of files and line numbers and lets you jump among them
- Usage: `:copen` and `:cclose`
  - Or make a toggle, see my vimrc at the end
- Load a file into it with `:cf /tmp/filename`
- Use `:cn` to jump to next fix (you should map this)
- Another way to populate it is the `:grep` command
- Can use anything that specifies line info; I use graudit with some custom regexes

Quickfix

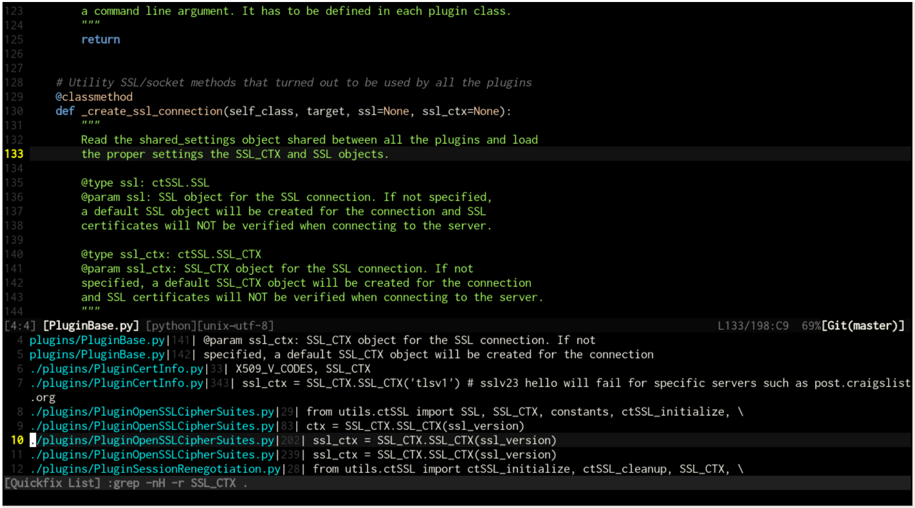
Using grep/vimgrep



```
from SSLSocket import SSLSocket


class HTTPSConnection(HTTPConnection):
    """
    This class mirrors httplib.HTTPSConnection but uses ctSSL instead of the
    standard ssl module.
    For now the way to access low level SSL functions associated with a given
    HTTPSConnection is to directly access the ssl and ssl_ctx attributes of the
    object. TODO: change that.

    @type ssl_ctx: ctSSL.SSL_CTX
    @ivar ssl_ctx: SSL_CTX object for the HTTPS connection.

    @type ssl: ctSSL.SSL
    @ivar ssl: SSL object for the HTTPS connection.
    certificates.
    """

    default_port = HTTPS_PORT

    def __init__(self, host, port=None, ssl=None, ssl_ctx=None,
[4:5] [HTTPSConnection.py] [python][unix-utf-8]                    L35/117:C1    32%[Git(master)]
  1
[Quickfix List]
:gr -r SSL_CTX .
```

Quickfix

Using grep/vimgrep



```
        a command line argument. It has to be defined in each plugin class.
        """
        return


    # Utility SSL/socket methods that turned out to be used by all the plugins
    @classmethod
    def _create_ssl_connection(self_class, target, ssl=None, ssl_ctx=None):
        """
        Read the shared_settings object shared between all the plugins and load
        the proper settings the SSL_CTX and SSL objects.

        @type ssl: ctSSL.SSL
        @param ssl: SSL object for the SSL connection. If not specified,
        a default SSL object will be created for the connection and SSL
        certificates will NOT be verified when connecting to the server.

        @type ssl_ctx: ctSSL.SSL_CTX
        @param ssl_ctx: SSL_CTX object for the SSL connection. If not
        specified, a default SSL_CTX object will be created for the connection
        and SSL certificates will NOT be verified when connecting to the server.
        """
[4:4] [PluginBase.py] [python][unix-utf-8]                                                  L133/198:C9  69%[Git(master)]
plugins/PluginBase.py|141| @param ssl_ctx: SSL_CTX object for the SSL connection. If not
plugins/PluginBase.py|142| specified, a default SSL_CTX object will be created for the connection
./plugins/PluginCertInfo.py|33| X509_V_CODES, SSL_CTX
./plugins/PluginCertInfo.py|343| ssl_ctx = SSL_CTX.SSL_CTX('tlsv1') # sslv23 hello will fail for specific servers such as post.craigslist
.org
./plugins/PluginOpenSSLCipherSuites.py|29| from utils.ctSSL import SSL, SSL_CTX, constants, ctSSL_initialize, \
./plugins/PluginOpenSSLCipherSuites.py|83| ctx = SSL_CTX.SSL_CTX(ssl_version)
./plugins/PluginOpenSSLCipherSuites.py|202| ssl_ctx = SSL_CTX.SSL_CTX(ssl_version)
./plugins/PluginOpenSSLCipherSuites.py|239| ssl_ctx = SSL_CTX.SSL_CTX(ssl_version)
./plugins/PluginSessionRenegotiation.py|28| from utils.ctSSL import ctSSL_initialize, ctSSL_cleanup, SSL_CTX, \
[Quickfix List] :grep -nH -r SSL_CTX .
```

File Browsing

netrw

- Use splits and the :Explore family
- Usage:
  - `:Explore` — open a file browser in your current window (if unmodified — otherwise, split first)
  - `:Sexplore` — open a browser in a split (horizontal)
  - `:Vexplore` — open a browser in a split (left-vertical)
  - And so on, `:help Explore`
- Use `mb` bookmark files for later examination or common use
- `qb` to query bookmarks
- See my .vimrc at the end for a function to toggle :Vexplore on and off like a file drawer
- You may want https://github.com/tpope/vim-vinegar

Plugin management
vim-plug

- There are several plugin managers
- The correct one is vim-plug:
  - https://github.com/junegunn/vim-plug

vim-plug install
```
curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
    https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
```

vim-plug vimrc
```
call plug#begin('~/.vim/plugged')
Plug 'Raimondi/delimitMate'
Plug 'SirVer/ultisnips' | Plug 'honza/vim-snippets'
call plug#end()
```

Quick file opening
FZF

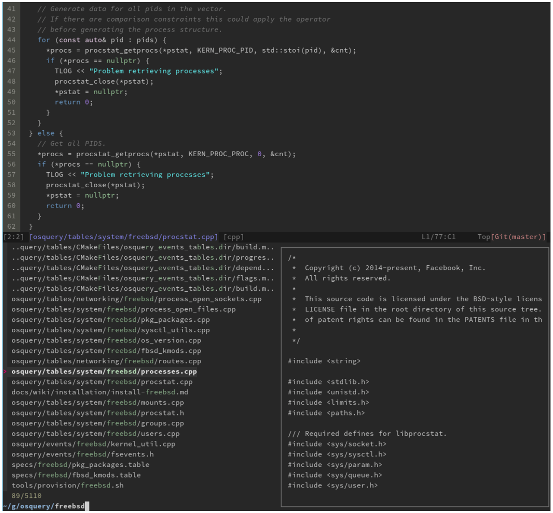
- Sometimes, you don’t want to browse, you want a certain file
- What if I have a giant codebase and don’t know where my file is?
- When you know what you want, regardless of where or what it is, use FZF:

https://github.com/junegunn/fzf https://github.com/junegunn/fzf.vim

- Can also search open buffers, recently used files, tags, etc.

Quick file opening

FZF in action



```
    // Generate data for all pids in the vector.
    // If there are comparison constraints this could apply the operator
    // before generating the process structure.
    for (const auto& pid : pids) {
      *procs = procstat_getprocs(*pstat, KERN_PROC_PID, std::stoi(pid), &cnt);
      if (*procs == nullptr) {
        TLOG << "Problem retrieving processes";
        procstat_close(*pstat);
        *pstat = nullptr;
        return 0;
      }
    }
  } else {
    // Get all PIDS.
    *procs = procstat_getprocs(*pstat, KERN_PROC_PROC, 0, &cnt);
    if (*procs == nullptr) {
      TLOG << "Problem retrieving processes";
      procstat_close(*pstat);
      *pstat = nullptr;
      return 0;
    }
  }
[2:2] [osquery/tables/system/freebsd/procstat.cpp] [cpp]          L1/77:C1     Top[Git(master)]
```

```
  ..query/tables/CMakeFiles/osquery_events_tables.dir/build.m..
  ..query/tables/CMakeFiles/osquery_events_tables.dir/progres..
  ..query/tables/CMakeFiles/osquery_events_tables.dir/depend...
  ..query/tables/CMakeFiles/osquery_events_tables.dir/flags.m..
  ..query/tables/CMakeFiles/osquery_events_tables.dir/build.m..
  osquery/tables/networking/freebsd/process_open_sockets.cpp
  osquery/tables/system/freebsd/process_open_files.cpp
  osquery/tables/system/freebsd/pkg_packages.cpp
  osquery/tables/system/freebsd/sysctl_utils.cpp
  osquery/tables/system/freebsd/os_version.cpp
  osquery/tables/system/freebsd/fbsd_kmods.cpp
  osquery/tables/networking/freebsd/routes.cpp
> osquery/tables/system/freebsd/processes.cpp
  osquery/tables/system/freebsd/procstat.cpp
  docs/wiki/installation/install-freebsd.md
  osquery/tables/system/freebsd/mounts.cpp
  osquery/tables/system/freebsd/procstat.h
  osquery/tables/system/freebsd/groups.cpp
  osquery/tables/system/freebsd/users.cpp
  osquery/events/freebsd/kernel_util.cpp
  osquery/events/freebsd/fsevents.h
  specs/freebsd/pkg_packages.table
  specs/freebsd/fbsd_kmods.table
  tools/provision/freebsd.sh
  89/5110
~/g/osquery/freebsd
```

```
/*
 *  Copyright (c) 2014-present, Facebook, Inc.
 *  All rights reserved.
 *
 *  This source code is licensed under the BSD-style licens
 *  LICENSE file in the root directory of this source tree.
 *  of patent rights can be found in the PATENTS file in th
 *
 */

#include <string>

#include <stdlib.h>
#include <unistd.h>
#include <limits.h>
#include <paths.h>

/// Required defines for libprocstat.
#include <sys/socket.h>
#include <sys/sysctl.h>
#include <sys/param.h>
#include <sys/queue.h>
#include <sys/user.h>
```

Exuberant Ctags

Tokenize all the things

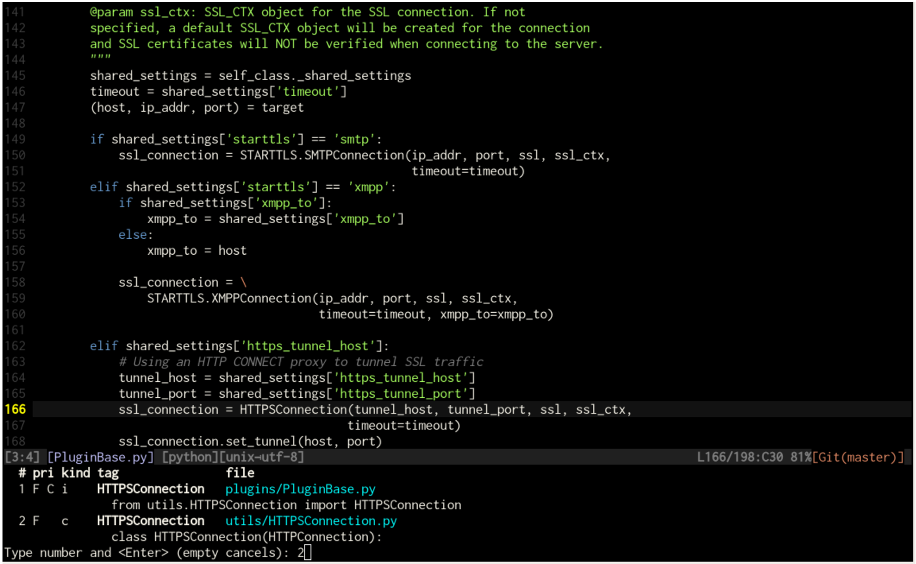
http://ctags.sourceforge.net/

- Generates an index of symbols
- Usage: at root of source tree (on command line), `ctags -R .`
- In vim, `:set tags=./tags` or whatever path you choose
- When cursor is on a function/method/whatever, `(CTRL-])` jumps to its definition
  - If ambiguous, a select list is displayed
- Return to your previous location with `(CTRL-t)`
- To open the tag in a "Preview" window, use `(CTRL-w ])`
  - Close with `(CTRL-w z)`
- You can also use FZF for this:

FZF tag jump

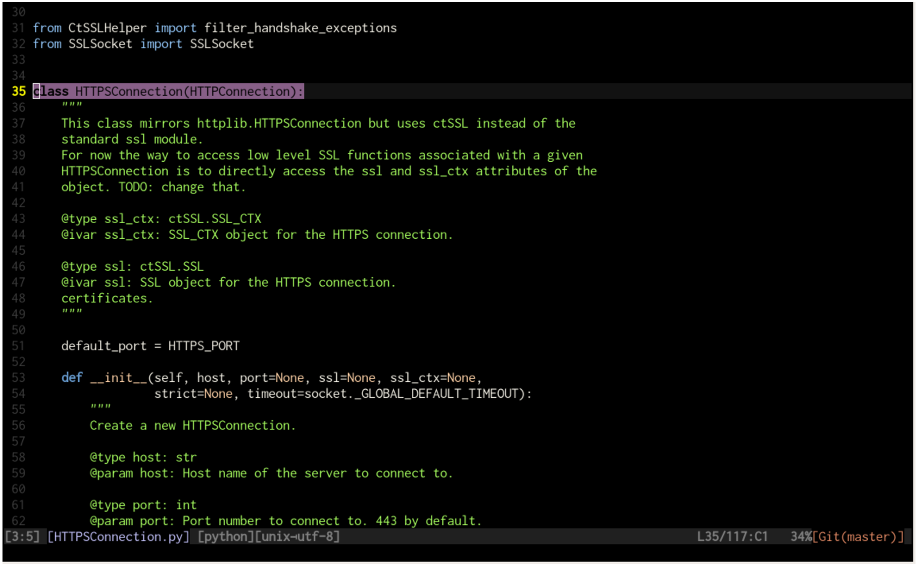
```
nmap <C-]> :call fzf#vim#tags(expand('<cword>'), \
    {'options': '--exact --select-1 --exit-0'})<CR>
```

Ctags usage: jump to definition



```
        @param ssl_ctx: SSL_CTX object for the SSL connection. If not
        specified, a default SSL_CTX object will be created for the connection
        and SSL certificates will NOT be verified when connecting to the server.
        """
        shared_settings = self_class._shared_settings
        timeout = shared_settings['timeout']
        (host, ip_addr, port) = target

        if shared_settings['starttls'] == 'smtp':
            ssl_connection = STARTTLS.SMTPConnection(ip_addr, port, ssl, ssl_ctx,
                                                     timeout=timeout)
        elif shared_settings['starttls'] == 'xmpp':
            if shared_settings['xmpp_to']:
                xmpp_to = shared_settings['xmpp_to']
            else:
                xmpp_to = host

            ssl_connection = \
                STARTTLS.XMPPConnection(ip_addr, port, ssl, ssl_ctx,
                                        timeout=timeout, xmpp_to=xmpp_to)

        elif shared_settings['https_tunnel_host']:
            # Using an HTTP CONNECT proxy to tunnel SSL traffic
            tunnel_host = shared_settings['https_tunnel_host']
            tunnel_port = shared_settings['https_tunnel_port']
            ssl_connection = HTTPSConnection(tunnel_host, tunnel_port, ssl, ssl_ctx,
                                             timeout=timeout)
            ssl_connection.set_tunnel(host, port)
[3:4] [PluginBase.py] [python][unix-utf-8]                                    L166/198:C30 81%[Git(master)]
  # pri kind tag               file
  1 F C i    HTTPSConnection   plugins/PluginBase.py
               from utils.HTTPSConnection import HTTPSConnection
  2 F   c    HTTPSConnection   utils/HTTPSConnection.py
               class HTTPSConnection(HTTPConnection):
Type number and <Enter> (empty cancels): 2
```

Ctags usage: successful symbol lookup



```
from CtSSLHelper import filter_handshake_exceptions
from SSLSocket import SSLSocket


class HTTPSConnection(HTTPConnection):
    """
    This class mirrors httplib.HTTPSConnection but uses ctSSL instead of the
    standard ssl module.
    For now the way to access low level SSL functions associated with a given
    HTTPSConnection is to directly access the ssl and ssl_ctx attributes of the
    object. TODO: change that.

    @type ssl_ctx: ctSSL.SSL_CTX
    @ivar ssl_ctx: SSL_CTX object for the HTTPS connection.

    @type ssl: ctSSL.SSL
    @ivar ssl: SSL object for the HTTPS connection.
    certificates.
    """

    default_port = HTTPS_PORT

    def __init__(self, host, port=None, ssl=None, ssl_ctx=None,
                 strict=None, timeout=socket._GLOBAL_DEFAULT_TIMEOUT):
        """
        Create a new HTTPSConnection.

        @type host: str
        @param host: Host name of the server to connect to.

        @type port: int
        @param port: Port number to connect to. 443 by default.
[3:5] [HTTPSConnection.py] [python][unix-utf-8]                         L35/117:C1    34%[Git(master)]
```

Tagbar

Intelligent symbol browsing

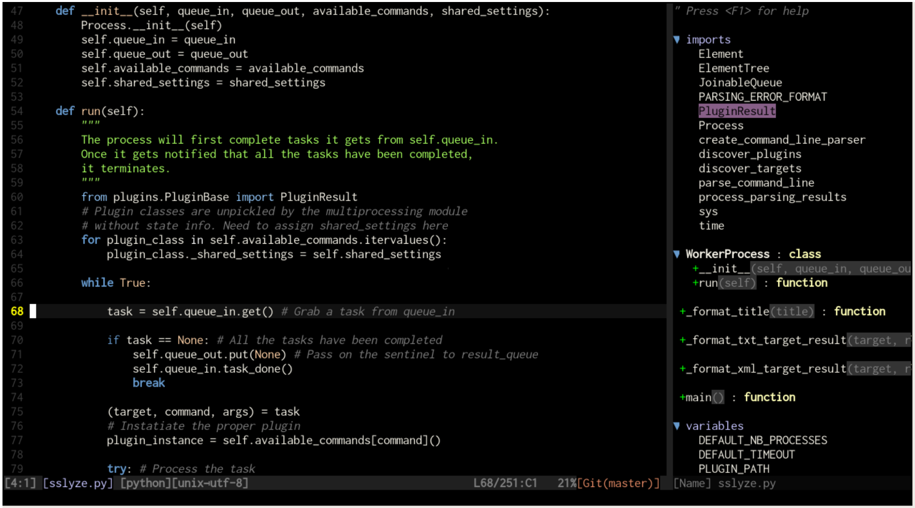
- Uses ctags to generate a symbol list
- Smart about identifying different symbol types
- Scope-aware
- Displays basic function signatures

TagBar mappings

```
map <silent> <F10> :TagbarToggle<CR>
nnoremap <silent> <F10> :TagbarToggle<CR>
```

Tagbar

Example Java tagbar



```
    def __init__(self, queue_in, queue_out, available_commands, shared_settings):
        Process.__init__(self)
        self.queue_in = queue_in
        self.queue_out = queue_out
        self.available_commands = available_commands
        self.shared_settings = shared_settings

    def run(self):
        """
        The process will first complete tasks it gets from self.queue_in.
        Once it gets notified that all the tasks have been completed,
        it terminates.
        """
        from plugins.PluginBase import PluginResult
        # Plugin classes are unpickled by the multiprocessing module
        # without state info. Need to assign shared_settings here
        for plugin_class in self.available_commands.itervalues():
            plugin_class._shared_settings = self.shared_settings

        while True:

            task = self.queue_in.get() # Grab a task from queue_in

            if task == None: # All the tasks have been completed
                self.queue_out.put(None) # Pass on the sentinel to result_queue
                self.queue_in.task_done()
                break

            (target, command, args) = task
            # Instatiate the proper plugin
            plugin_instance = self.available_commands[command]()

            try: # Process the task
[4:1] [sslyze.py] [python][unix-utf-8]               L68/251:C1    21%[Git(master)]
```

```
" Press <F1> for help

▼ imports
    Element
    ElementTree
    JoinableQueue
    PARSING_ERROR_FORMAT
    PluginResult
    Process
    create_command_line_parser
    discover_plugins
    discover_targets
    parse_command_line
    process_parsing_results
    sys
    time

▼ WorkerProcess : class
   +__init__(self, queue_in, queue_ou
   +run(self) : function

 +_format_title(title) : function

 +_format_txt_target_result(target, r

 +_format_xml_target_result(target, r

 +main() : function

▼ variables
    DEFAULT_NB_PROCESSES
    DEFAULT_TIMEOUT
    PLUGIN_PATH
[Name] sslyze.py
```

Cscope

Serious symbol mangling

http://cscope.sourceforge.net/cscope_vim_tutorial.html

- Similar to ctags, but lets you do much more:
  - Show all callers of a function
  - Show all instances of a tag
  - Open include files
- Usage: `find . -name "*.[ch]" > cscope.files && cscope -b`
- Or whatever type of file you want to index

Cscope
Cheat sheet

- I recommend the plugin from brookhong/cscope.vim
- Query a symbol with `(\)` followed by one of the below:

```
cscope query types
"   's'   symbol: find all references to the token under cursor
"   'g'   global: find global definition(s) of the token under cursor
"   'c'   calls:  find all calls to the function name under cursor
"   't'   text:   find all instances of the text under cursor
"   'e'   egrep:  egrep search for the word under cursor
"   'f'   file:   open the filename under cursor
"   'i'   includes: find files that include the filename under cursor
"   'd'   called: find functions that function under cursor calls
```

Cscope
Configuration

- By default, a numbered list pops up, similar to multiple matches for ctags
- I prefer to have most of them be in the quickfix list, so:

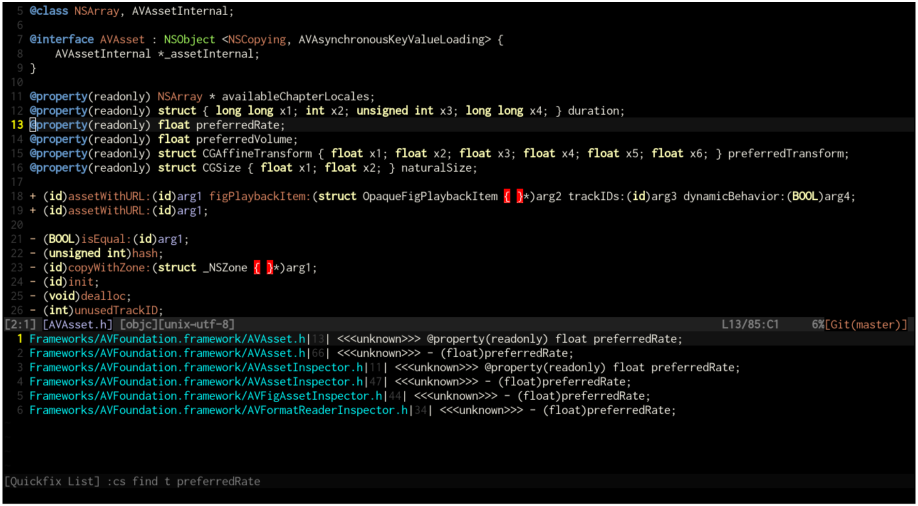
cscope quickfix types

```
    set cscopequickfix=s-,c-,d-,i-,t-,e-
```

- "-" means to make a new quickfix list, "+" means to append

Cscope

Querytype “t” in the quickfix



```
@class NSArray, AVAssetInternal;

@interface AVAsset : NSObject <NSCopying, AVAsynchronousKeyValueLoading> {
    AVAssetInternal *_assetInternal;
}

@property(readonly) NSArray * availableChapterLocales;
@property(readonly) struct { long long x1; int x2; unsigned int x3; long long x4; } duration;
@property(readonly) float preferredRate;
@property(readonly) float preferredVolume;
@property(readonly) struct CGAffineTransform { float x1; float x2; float x3; float x4; float x5; float x6; } preferredTransform;
@property(readonly) struct CGSize { float x1; float x2; } naturalSize;

+ (id)assetWithURL:(id)arg1 figPlaybackItem:(struct OpaqueFigPlaybackItem { }*)arg2 trackIDs:(id)arg3 dynamicBehavior:(BOOL)arg4;
+ (id)assetWithURL:(id)arg1;

- (BOOL)isEqual:(id)arg1;
- (unsigned int)hash;
- (id)copyWithZone:(struct _NSZone { }*)arg1;
- (id)init;
- (void)dealloc;
- (int)unusedTrackID;
[2:1] [AVAsset.h] [objc][unix-utf-8]                                    L13/85:C1     6%[Git(master)]
Frameworks/AVFoundation.framework/AVAsset.h|13| <<<unknown>>> @property(readonly) float preferredRate;
Frameworks/AVFoundation.framework/AVAsset.h|66| <<<unknown>>> - (float)preferredRate;
Frameworks/AVFoundation.framework/AVAssetInspector.h|11| <<<unknown>>> @property(readonly) float preferredRate;
Frameworks/AVFoundation.framework/AVAssetInspector.h|47| <<<unknown>>> - (float)preferredRate;
Frameworks/AVFoundation.framework/AVFigAssetInspector.h|44| <<<unknown>>> - (float)preferredRate;
Frameworks/AVFoundation.framework/AVFormatReaderInspector.h|34| <<<unknown>>> - (float)preferredRate;
~
~
~
[Quickfix List] :cs find t preferredRate
```

Completion

Cruise control for completion

- Vim's default code completion mechanism is rather complex
- Ties hands into knots, requires multiple keystrokes
- Worth looking into for some things, e.g. `CTRL-X CTRL-L`
- Lots of completion mechanisms available, some very heavy, some very light
- VimCompletesMe makes it just work with tab
    - https://github.com/ajh17/VimCompletesMe/
- Maybe also check out deoplete, YCM

Snippets

You will forget how to program forever

- Spits out code snippets from abbreviations
- Easy to define new ones
- Several options, I suggest: `https://github.com/SirVer/ultisnips`
- Note that the default binding is tab, conflicts with tab-completion plugins (VCM)
  - Remap to something else

Snippets

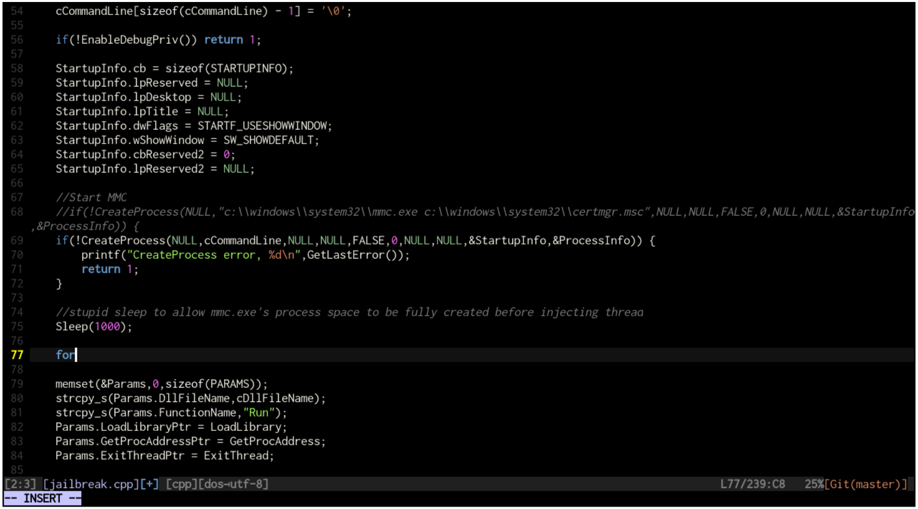
Example snippets

tex.snippets

```
snippet fig
\begin{figure}[ht]
    \center \includegraphics[width=1.0\textwidth]{images/${1}}
    \caption{${2}}
\end{figure}

snippet lst
\begin{lstlisting}[style=code,language=${1},numbers=none,caption={${2}}]

\end{lstlisting}
```

Snippets

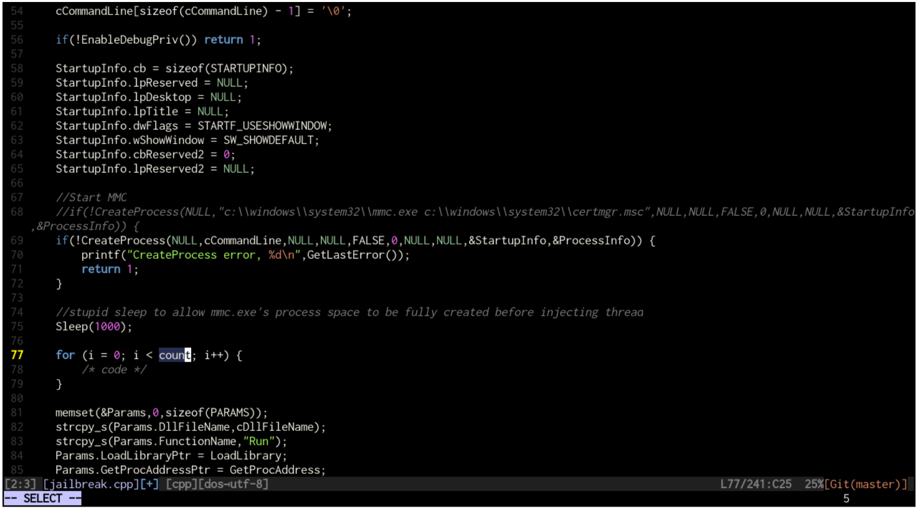
Complete a “for” loop



```
    cCommandLine[sizeof(cCommandLine) - 1] = '\0';

    if(!EnableDebugPriv()) return 1;

    StartupInfo.cb = sizeof(STARTUPINFO);
    StartupInfo.lpReserved = NULL;
    StartupInfo.lpDesktop = NULL;
    StartupInfo.lpTitle = NULL;
    StartupInfo.dwFlags = STARTF_USESHOWWINDOW;
    StartupInfo.wShowWindow = SW_SHOWDEFAULT;
    StartupInfo.cbReserved2 = 0;
    StartupInfo.lpReserved2 = NULL;

    //Start MMC
    //if(!CreateProcess(NULL,"c:\\windows\\system32\\mmc.exe c:\\windows\\system32\\certmgr.msc",NULL,NULL,FALSE,0,NULL,NULL,&StartupInfo
,&ProcessInfo)) {
    if(!CreateProcess(NULL,cCommandLine,NULL,NULL,FALSE,0,NULL,NULL,&StartupInfo,&ProcessInfo)) {
        printf("CreateProcess error, %d\n",GetLastError());
        return 1;
    }

    //stupid sleep to allow mmc.exe's process space to be fully created before injecting thread
    Sleep(1000);

    for

    memset(&Params,0,sizeof(PARAMS));
    strcpy_s(Params.DllFileName,cDllFileName);
    strcpy_s(Params.FunctionName,"Run");
    Params.LoadLibraryPtr = LoadLibrary;
    Params.GetProcAddressPtr = GetProcAddress;
    Params.ExitThreadPtr = ExitThread;

[2:3] [jailbreak.cpp][+] [cpp][dos-utf-8]                                    L77/239:C8    25%[Git(master)]
-- INSERT --
```

Snippets

Snippet success



```
    cCommandLine[sizeof(cCommandLine) - 1] = '\0';

    if(!EnableDebugPriv()) return 1;

    StartupInfo.cb = sizeof(STARTUPINFO);
    StartupInfo.lpReserved = NULL;
    StartupInfo.lpDesktop = NULL;
    StartupInfo.lpTitle = NULL;
    StartupInfo.dwFlags = STARTF_USESHOWWINDOW;
    StartupInfo.wShowWindow = SW_SHOWDEFAULT;
    StartupInfo.cbReserved2 = 0;
    StartupInfo.lpReserved2 = NULL;

    //Start MMC
    //if(!CreateProcess(NULL,"c:\\windows\\system32\\mmc.exe c:\\windows\\system32\\certmgr.msc",NULL,NULL,FALSE,0,NULL,NULL,&StartupInfo
,&ProcessInfo)) {
    if(!CreateProcess(NULL,cCommandLine,NULL,NULL,FALSE,0,NULL,NULL,&StartupInfo,&ProcessInfo)) {
        printf("CreateProcess error, %d\n",GetLastError());
        return 1;
    }

    //stupid sleep to allow mmc.exe's process space to be fully created before injecting thread
    Sleep(1000);

    for (i = 0; i < count; i++) {
        /* code */
    }

    memset(&Params,0,sizeof(PARAMS));
    strcpy_s(Params.DllFileName,cDllFileName);
    strcpy_s(Params.FunctionName,"Run");
    Params.LoadLibraryPtr = LoadLibrary;
    Params.GetProcAddressPtr = GetProcAddress;
[2:3] [jailbreak.cpp][+] [cpp][dos-utf-8]                                        L77/241:C25  25%[Git(master)]
-- SELECT --                                                                            5
```

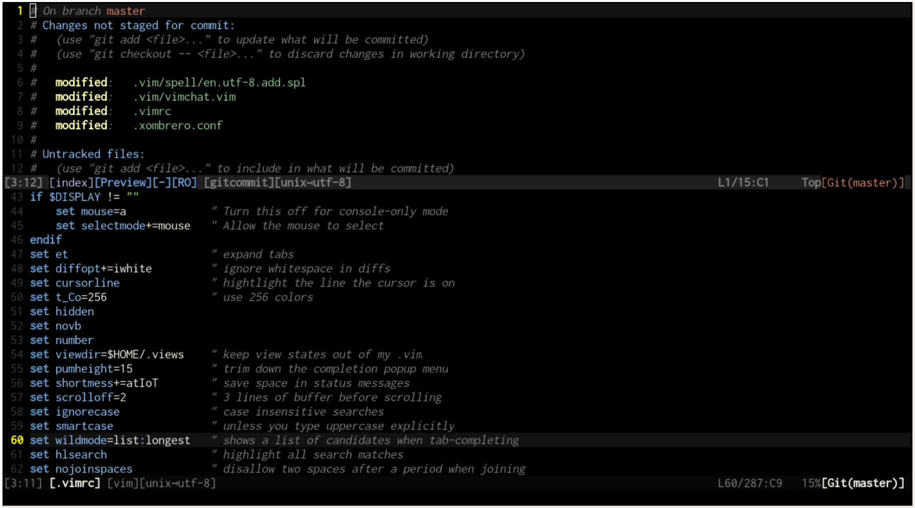
Fugitive

git in vim

- Do all your git magic from within vim
- Often way nicer than command line
- Navigate diffs
- Check in/check out/stage/commit
- Check out: `http://vimcasts.org/episodes`
- Short story: `:Gw`, `:Gr`, `:Gstatus`, `:Gcommit`, `:Gdiff`

Fugitive

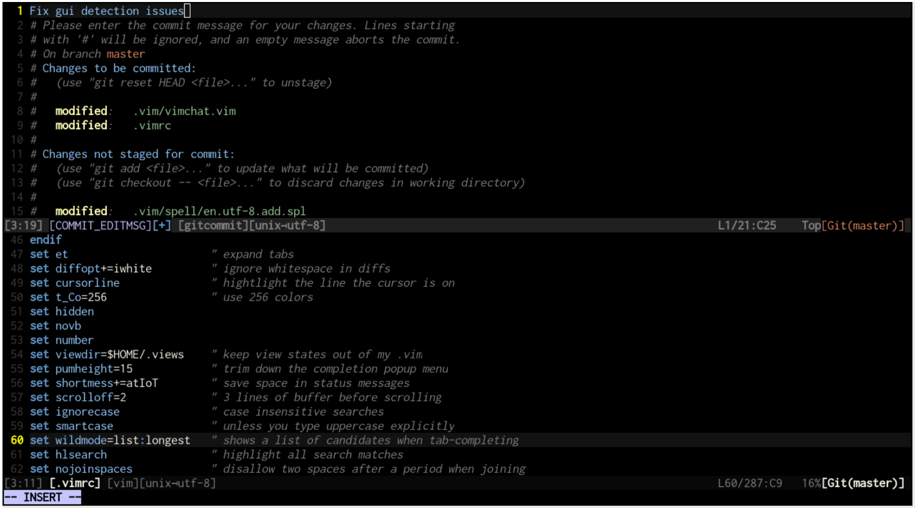
:Gstatus



```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#	modified:   .vim/spell/en.utf-8.add.spl
#	modified:   .vim/vimchat.vim
#	modified:   .vimrc
#	modified:   .xombrero.conf
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
[3:12] [index][Preview][-][RO] [gitcommit][unix-utf-8]                L1/15:C1     Top[Git(master)]
if $DISPLAY != ""
    set mouse=a                 " Turn this off for console-only mode
    set selectmode+=mouse       " Allow the mouse to select
endif
set et                          " expand tabs
set diffopt+=iwhite             " ignore whitespace in diffs
set cursorline                  " hightlight the line the cursor is on
set t_Co=256                    " use 256 colors
set hidden
set novb
set number
set viewdir=$HOME/.views        " keep view states out of my .vim
set pumheight=15                " trim down the completion popup menu
set shortmess+=atIoT            " save space in status messages
set scrolloff=2                 " 3 lines of buffer before scrolling
set ignorecase                  " case insensitive searches
set smartcase                   " unless you type uppercase explicitly
set wildmode=list:longest       " shows a list of candidates when tab-completing
set hlsearch                    " highlight all search matches
set nojoinspaces                " disallow two spaces after a period when joining
[3:11] [.vimrc] [vim][unix-utf-8]                                     L60/287:C9   15%[Git(master)]
```

Fugitive

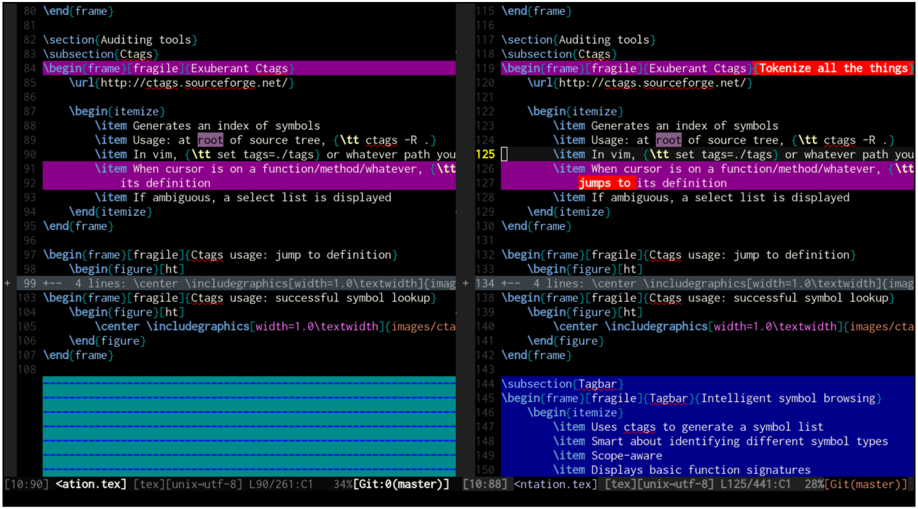
:Gcommit



```
Fix gui detection issues
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#    (use "git reset HEAD <file>..." to unstage)
#
#    modified:   .vim/vimchat.vim
#    modified:   .vimrc
#
# Changes not staged for commit:
#    (use "git add <file>..." to update what will be committed)
#    (use "git checkout -- <file>..." to discard changes in working directory)
#
#    modified:   .vim/spell/en.utf-8.add.spl
[3:19] [COMMIT_EDITMSG][+] [gitcommit][unix-utf-8]                                          L1/21:C25     Top[Git(master)]
endif
set et                         " expand tabs
set diffopt+=iwhite            " ignore whitespace in diffs
set cursorline                 " hightlight the line the cursor is on
set t_Co=256                   " use 256 colors
set hidden
set novb
set number
set viewdir=$HOME/.views       " keep view states out of my .vim
set pumheight=15               " trim down the completion popup menu
set shortmess+=atIoT           " save space in status messages
set scrolloff=2                " 3 lines of buffer before scrolling
set ignorecase                 " case insensitive searches
set smartcase                  " unless you type uppercase explicitly
set wildmode=list:longest      " shows a list of candidates when tab-completing
set hlsearch                   " highlight all search matches
set nojoinspaces               " disallow two spaces after a period when joining
[3:11] [.vimrc] [vim][unix-utf-8]                                                           L60/287:C9    16%[Git(master)]
-- INSERT --
```

Fugitive

:Gdiff



```
\end{frame}

\section{Auditing tools}
\subsection{Ctags}
\begin{frame}[fragile]{Exuberant Ctags}
    \url{http://ctags.sourceforge.net/}

    \begin{itemize}
        \item Generates an index of symbols
        \item Usage: at root of source tree, {\tt ctags -R .}
        \item In vim, {\tt set tags=./tags} or whatever path you
        \item When cursor is on a function/method/whatever, {\tt
            its definition
        \item If ambiguous, a select list is displayed
    \end{itemize}
\end{frame}

\begin{frame}[fragile]{Ctags usage: jump to definition}
    \begin{figure}[ht]
+ +--- 4 lines: \center \includegraphics[width=1.0\textwidth]{imag
\begin{frame}[fragile]{Ctags usage: successful symbol lookup}
    \begin{figure}[ht]
        \center \includegraphics[width=1.0\textwidth]{images/cta
    \end{figure}
\end{frame}

[10:90] <ation.tex] [tex][unix-utf-8] L90/261:C1    34%[Git:0(master)]
```

```
\end{frame}

\section{Auditing tools}
\subsection{Ctags}
\begin{frame}[fragile]{Exuberant Ctags}{Tokenize all the things}
    \url{http://ctags.sourceforge.net/}

    \begin{itemize}
        \item Generates an index of symbols
        \item Usage: at root of source tree, {\tt ctags -R .}
        \item In vim, {\tt set tags=./tags} or whatever path you
        \item When cursor is on a function/method/whatever, {\tt
            jumps to its definition
        \item If ambiguous, a select list is displayed
    \end{itemize}
\end{frame}

\begin{frame}[fragile]{Ctags usage: jump to definition}
    \begin{figure}[ht]
+ +--- 4 lines: \center \includegraphics[width=1.0\textwidth]{imag
\begin{frame}[fragile]{Ctags usage: successful symbol lookup}
    \begin{figure}[ht]
        \center \includegraphics[width=1.0\textwidth]{images/cta
    \end{figure}
\end{frame}

\subsection{Tagbar}
\begin{frame}[fragile]{Tagbar}{Intelligent symbol browsing}
    \begin{itemize}
        \item Uses ctags to generate a symbol list
        \item Smart about identifying different symbol types
        \item Scope-aware
        \item Displays basic function signatures
[10:88] <ntation.tex] [tex][unix-utf-8] L125/441:C1   28%[Git(master)]
```

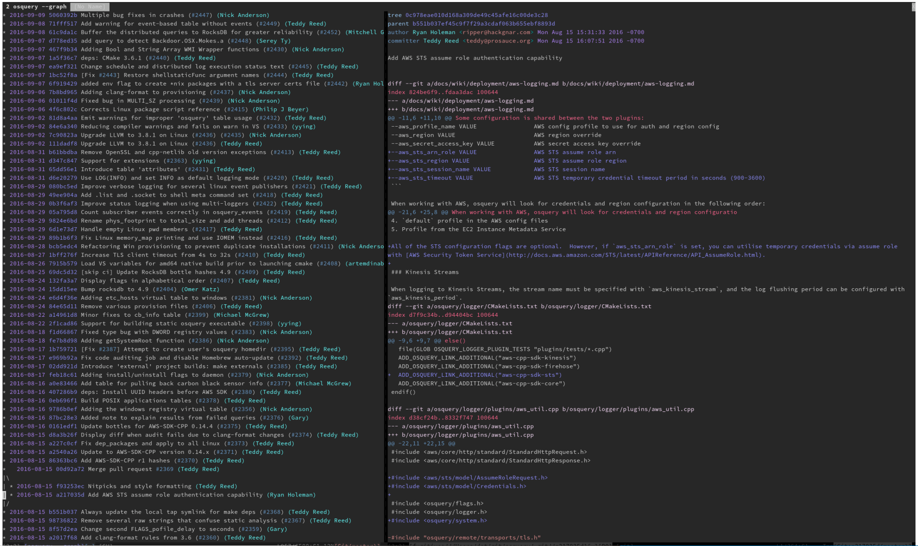
gv

Your non-GUI git GUI

- Plugin for Fugitive
- Commit browsing
- Perusing file history
- Jumping into vimdiff
- https://github.com/junegunn/gv.vim

gv

Browser mode



```
2 osquery --graph  [No Name]
* 2016-09-09 5060392b Multiple bug fixes in crashes (#2447) (Nick Anderson)
* 2016-09-08 71fff517 Add warning for event-based table without events (#2449) (Teddy Reed)
* 2016-09-08 61c9da1c Buffer the distributed queries to RocksDB for greater reliability (#2452) (Mitchell G
* 2016-09-07 d778ed35 add query to detect Backdoor.OSX.Mokes.a (#2448) (Serey Ty)
* 2016-09-07 467f9b34 Adding Bool and String Array WMI Wrapper functions (#2430) (Nick Anderson)
* 2016-09-07 1a5f36c7 deps: CMake 3.6.1 (#2440) (Teddy Reed)
* 2016-09-07 ea9ef321 Change schedule and distributed log execution status text (#2445) (Teddy Reed)
* 2016-09-07 1bc52f8a [Fix #2443] Restore shellstaticFunc argument names (#2444) (Teddy Reed)
* 2016-09-07 6f919429 added env flag to create *nix packages with a tls server certs file (#2442) (Ryan Hol
* 2016-09-06 7b8bd965 Adding clang-format to provisioning (#2437) (Nick Anderson)
* 2016-09-06 01011f4d Fixed bug in MULTI_SZ processing (#2439) (Nick Anderson)
* 2016-09-06 4f6c802c Corrects Linux package script reference (#2415) (Philip J Beyer)
* 2016-09-02 81d8a4aa Emit warnings for improper 'osquery' table usage (#2432) (Teddy Reed)
* 2016-09-02 84e6a340 Reducing compiler warnings and fails on warn in VS (#2433) (yying)
* 2016-09-02 7c90823a Upgrade LLVM to 3.8.1 on Linux (#2436) (#2435) (Nick Anderson)
* 2016-09-02 111dadf8 Upgrade LLVM to 3.8.1 on Linux (#2436) (Teddy Reed)
* 2016-08-31 b61bbdba Remove OpenSSL and cpp-netlib old version exceptions (#2413) (Teddy Reed)
* 2016-08-31 d347c847 Support for extensions (#2363) (yying)
* 2016-08-31 65dd56e1 Introduce table 'attributes' (#2431) (Teddy Reed)
* 2016-08-31 d6e20279 Use LOG(INFO) and set INFO as default logging mode (#2420) (Teddy Reed)
* 2016-08-29 080bc5ed Improve verbose logging for several linux event publishers (#2421) (Teddy Reed)
* 2016-08-29 49ee904a Add .list and .socket to shell meta command set (#2418) (Teddy Reed)
* 2016-08-29 0b3f6af3 Improve status logging when using multi-loggers (#2422) (Teddy Reed)
* 2016-08-29 05a795d8 Count subscriber events correctly in osquery_events (#2419) (Teddy Reed)
* 2016-08-29 9824e6bd Rename phys_footprint to total_size and add threads (#2412) (Teddy Reed)
* 2016-08-29 6d1e73d7 Handle empty Linux pwd members (#2417) (Teddy Reed)
* 2016-08-29 89b1b6f3 Fix Linux memory_map printing and use IOMEM instead (#2416) (Teddy Reed)
* 2016-08-28 bcb5edc4 Refactoring Win provisioning to prevent duplicate installations (#2411) (Nick Anderso
* 2016-08-27 1bff276f Increase TLS client timeout from 4s to 32s (#2410) (Teddy Reed)
* 2016-08-26 7915b579 Load VS variables for amd64 native build prior to launching cmake (#2408) (artemdinab
* 2016-08-25 69dc5d32 [skip ci] Update RocksDB bottle hashes 4.9 (#2409) (Teddy Reed)
* 2016-08-24 132fa3a7 Display flags in alphabetical order (#2407) (Teddy Reed)
* 2016-08-24 15dd15ee Bump rocksdb to 4.9 (#2404) (Omer Katz)
* 2016-08-24 e6d4f36e Adding etc_hosts virtual table to windows (#2381) (Nick Anderson)
* 2016-08-24 84e65d11 Remove various provision files (#2406) (Teddy Reed)
* 2016-08-22 a14961d8 Minor fixes to cb_info table (#2399) (Michael McGrew)
* 2016-08-22 2f1cad86 Support for building static osquery executable (#2398) (yying)
* 2016-08-18 f1d66867 Fixed type bug with DWORD registry values (#2383) (Nick Anderson)
* 2016-08-18 fe7b8d98 Adding getSystemRoot function (#2386) (Nick Anderson)
* 2016-08-17 1b759721 [Fix #2387] Attempt to create user's osquery homedir (#2395) (Teddy Reed)
* 2016-08-17 e969b92a Fix code auditing job and disable Homebrew auto-update (#2392) (Teddy Reed)
* 2016-08-17 02dd921d Introduce 'external' project builds: make externals (#2385) (Teddy Reed)
* 2016-08-17 feb18c61 Adding install/uninstall flags to daemon (#2379) (Nick Anderson)
* 2016-08-16 a0e83466 Add table for pulling back carbon black sensor info (#2377) (Michael McGrew)
* 2016-08-16 407286b9 deps: Install UUID headers before AWS SDK (#2380) (Teddy Reed)
* 2016-08-16 0eb696f1 Build POSIX applications tables (#2378) (Teddy Reed)
* 2016-08-16 9786b0ef Adding the windows registry virtual table (#2356) (Nick Anderson)
* 2016-08-16 87bc28e3 Added note to explain results from failed queries (#2376) (Gary)
* 2016-08-16 0161edf1 Update bottles for AWS-SDK-CPP 0.14.4 (#2375) (Teddy Reed)
* 2016-08-15 d8a3b26f Display diff when audit fails due to clang-format changes (#2374) (Teddy Reed)
* 2016-08-15 a227c0cf Fix dep_packages and apply to all Linux (#2373) (Teddy Reed)
* 2016-08-15 a2540a26 Update to AWS-SDK-CPP version 0.14.x (#2371) (Teddy Reed)
* 2016-08-15 86363bc6 Add AWS-SDK-CPP r1 hashes (#2370) (Teddy Reed)
*   2016-08-15 00d92a72 Merge pull request #2369 (Teddy Reed)
|\
| * 2016-08-15 f93253ec Nitpicks and style formatting (Teddy Reed)
| * 2016-08-15 a217035d Add AWS STS assume role authentication capability (Ryan Holeman)
|/
* 2016-08-15 b551b037 Always update the local tap symlink for make deps (#2368) (Teddy Reed)
* 2016-08-15 98736822 Remove several raw strings that confuse static analysis (#2367) (Teddy Reed)
* 2016-08-15 8f57d2ea Change second FLAGS_pofile_delay to seconds (#2359) (Gary)
* 2016-08-15 a2017f68 Add clang-format rules from 3.6 (#2360) (Teddy Reed)
```

```
tree 0c978eae010d168a309de49c45afe16c00de3c28
parent b551b037ef45c9f7f29a3cdaf063b655ebf8893d
author Ryan Holeman <ripper@hackgnar.com> Mon Aug 15 15:31:33 2016 -0700
committer Teddy Reed <teddy@prosauce.org> Mon Aug 15 16:07:51 2016 -0700

Add AWS STS assume role authentication capability


diff --git a/docs/wiki/deployment/aws-logging.md b/docs/wiki/deployment/aws-logging.md
index 824be6f9..fdaa3dac 100644
--- a/docs/wiki/deployment/aws-logging.md
+++ b/docs/wiki/deployment/aws-logging.md
@@ -11,6 +11,10 @@ Some configuration is shared between the two plugins:
 --aws_profile_name VALUE                AWS config profile to use for auth and region config
 --aws_region VALUE                      AWS region override
 --aws_secret_access_key VALUE           AWS secret access key override
+--aws_sts_arn_role VALUE                AWS STS assume role arn
+--aws_sts_region VALUE                  AWS STS assume role region
+--aws_sts_session_name VALUE            AWS STS session name
+--aws_sts_timeout VALUE                 AWS STS temporary credential timeout period in seconds (900-3600)
 ```

 When working with AWS, osquery will look for credentials and region configuration in the following order:
@@ -21,6 +25,8 @@ When working with AWS, osquery will look for credentials and region configuratio
 4. `default` profile in the AWS config files
 5. Profile from the EC2 Instance Metadata Service

+All of the STS configuration flags are optional. However, if `aws_sts_arn_role` is set, you can utilise temporary credentials via assume role
with [AWS Security Token Service](http://docs.aws.amazon.com/STS/latest/APIReference/API_AssumeRole.html).
+
 ### Kinesis Streams

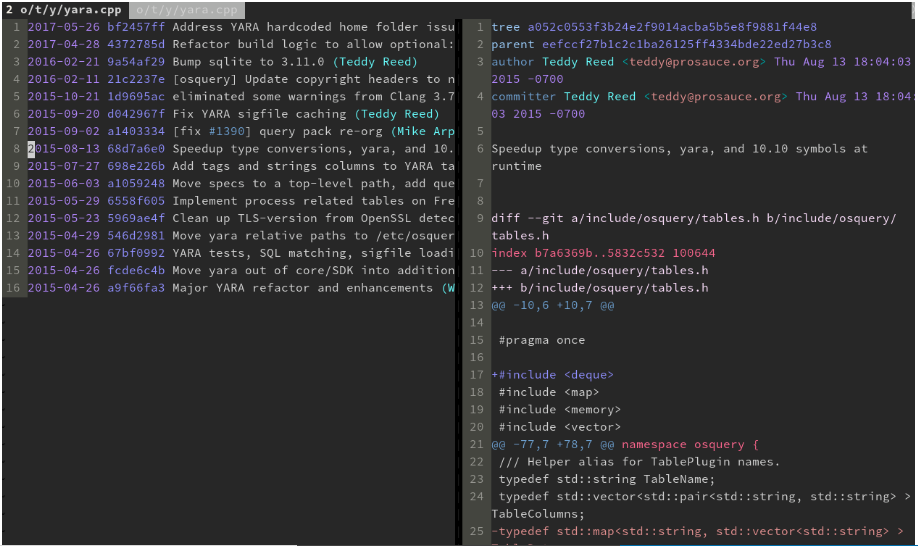
 When logging to Kinesis Streams, the stream name must be specified with `aws_kinesis_stream`, and the log flushing period can be configured with
`aws_kinesis_period`.
diff --git a/osquery/logger/CMakeLists.txt b/osquery/logger/CMakeLists.txt
index d7f9c34b..d94404bc 100644
--- a/osquery/logger/CMakeLists.txt
+++ b/osquery/logger/CMakeLists.txt
@@ -9,6 +9,7 @@ else()
   file(GLOB OSQUERY_LOGGER_PLUGIN_TESTS "plugins/tests/*.cpp")
   ADD_OSQUERY_LINK_ADDITIONAL("aws-cpp-sdk-kinesis")
   ADD_OSQUERY_LINK_ADDITIONAL("aws-cpp-sdk-firehose")
+ ADD_OSQUERY_LINK_ADDITIONAL("aws-cpp-sdk-sts")
   ADD_OSQUERY_LINK_ADDITIONAL("aws-cpp-sdk-core")
 endif()

diff --git a/osquery/logger/plugins/aws_util.cpp b/osquery/logger/plugins/aws_util.cpp
index d38cf24b..8332f747 100644
--- a/osquery/logger/plugins/aws_util.cpp
+++ b/osquery/logger/plugins/aws_util.cpp
@@ -22,11 +22,15 @@
 #include <aws/core/http/standard/StandardHttpRequest.h>
 #include <aws/core/http/standard/StandardHttpResponse.h>

+#include <aws/sts/model/AssumeRoleRequest.h>
+#include <aws/sts/model/Credentials.h>
+
 #include <osquery/flags.h>
 #include <osquery/logger.h>
+#include <osquery/system.h>

-#include "osquery/remote/transports/tls.h"
```
```

gv
File mode



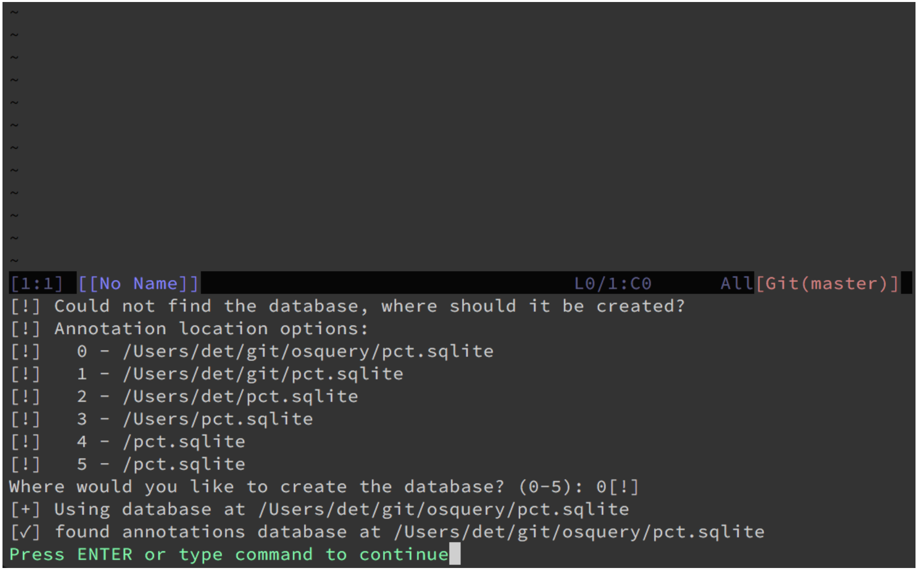
```
2 o/t/y/yara.cpp  o/t/y/yara.cpp
2017-05-26 bf2457ff Address YARA hardcoded home folder issu
2017-04-28 4372785d Refactor build logic to allow optional:
2016-02-21 9a54af29 Bump sqlite to 3.11.0 (Teddy Reed)
2016-02-11 21c2237e [osquery] Update copyright headers to n
2015-10-21 1d9695ac eliminated some warnings from Clang 3.7
2015-09-20 d042967f Fix YARA sigfile caching (Teddy Reed)
2015-09-02 a1403334 [fix #1390] query pack re-org (Mike Arp
2015-08-13 68d7a6e0 Speedup type conversions, yara, and 10.
2015-07-27 698e226b Add tags and strings columns to YARA ta
2015-06-03 a1059248 Move specs to a top-level path, add que
2015-05-29 6558f605 Implement process related tables on Fre
2015-05-23 5969ae4f Clean up TLS-version from OpenSSL detec
2015-04-29 546d2981 Move yara relative paths to /etc/osquer
2015-04-26 67bf0992 YARA tests, SQL matching, sigfile loadi
2015-04-26 fcde6c4b Move yara out of core/SDK into addition
2015-04-26 a9f66fa3 Major YARA refactor and enhancements (W
```

```
tree a052c0553f3b24e2f9014acba5b5e8f9881f44e8
parent eefccf27b1c2c1ba26125ff4334bde22ed27b3c8
author Teddy Reed <teddy@prosauce.org> Thu Aug 13 18:04:03
2015 -0700
committer Teddy Reed <teddy@prosauce.org> Thu Aug 13 18:04:
03 2015 -0700

Speedup type conversions, yara, and 10.10 symbols at
runtime


diff --git a/include/osquery/tables.h b/include/osquery/
tables.h
index b7a6369b..5832c532 100644
--- a/include/osquery/tables.h
+++ b/include/osquery/tables.h
@@ -10,6 +10,7 @@

 #pragma once

+#include <deque>
 #include <map>
 #include <memory>
 #include <vector>
@@ -77,7 +78,7 @@ namespace osquery {
 /// Helper alias for TablePlugin names.
 typedef std::string TableName;
 typedef std::vector<std::pair<std::string, std::string> >
TableColumns;
-typedef std::map<std::string, std::vector<std::string> >
```

Precise Code Tracking

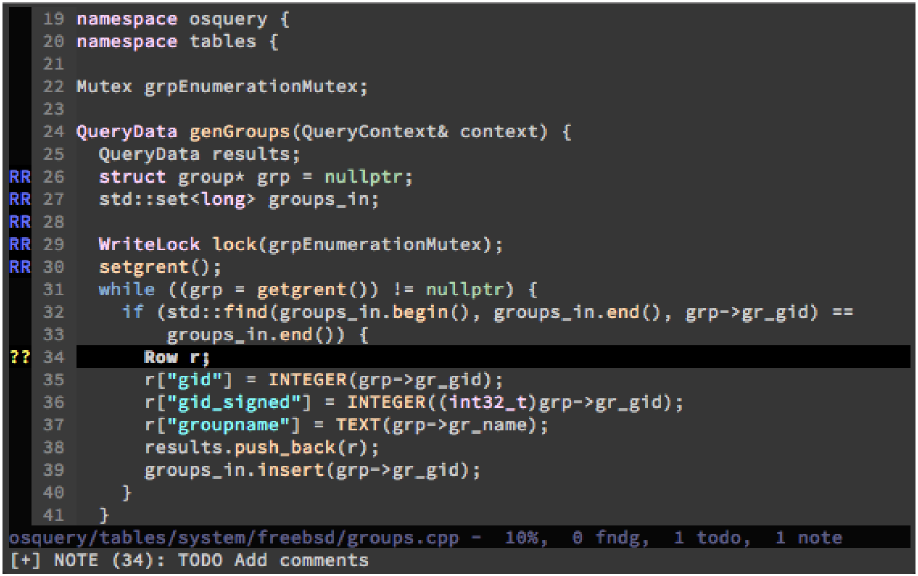
- Precise code tracking: `https://github.com/d0c-s4vage/pct-vim`
- Add todos, findings, annotations, or mark reviewed
- Show a report of all of the above

Precise Code Tracking



```
~
~
~
~
~
~
~
~
~
~
~
~
[1:1] [[No Name]]                                L0/1:C0      All[Git(master)]
[!] Could not find the database, where should it be created?
[!] Annotation location options:
[!]   0 - /Users/det/git/osquery/pct.sqlite
[!]   1 - /Users/det/git/pct.sqlite
[!]   2 - /Users/det/pct.sqlite
[!]   3 - /Users/pct.sqlite
[!]   4 - /pct.sqlite
[!]   5 - /pct.sqlite
Where would you like to create the database? (0-5): 0[!]
[+] Using database at /Users/det/git/osquery/pct.sqlite
[✓] found annotations database at /Users/det/git/osquery/pct.sqlite
Press ENTER or type command to continue
```

Precise Code Tracking



```
   19 namespace osquery {
   20 namespace tables {
   21
   22 Mutex grpEnumerationMutex;
   23
   24 QueryData genGroups(QueryContext& context) {
   25   QueryData results;
RR 26   struct group* grp = nullptr;
RR 27   std::set<long> groups_in;
RR 28
RR 29   WriteLock lock(grpEnumerationMutex);
RR 30   setgrent();
   31   while ((grp = getgrent()) != nullptr) {
   32     if (std::find(groups_in.begin(), groups_in.end(), grp->gr_gid) ==
   33         groups_in.end()) {
?? 34       Row r;
   35       r["gid"] = INTEGER(grp->gr_gid);
   36       r["gid_signed"] = INTEGER((int32_t)grp->gr_gid);
   37       r["groupname"] = TEXT(grp->gr_name);
   38       results.push_back(r);
   39       groups_in.insert(grp->gr_gid);
   40     }
   41   }
osquery/tables/system/freebsd/groups.cpp -  10%,  0 fndg,  1 todo,  1 note
[+] NOTE (34): TODO Add comments
```

Suggested Reading

- :help jump
- :help buffer
- :help window-move-cursor
- :help folds

Questions?

Check out my setup at

https://github.com/lxcode/dotfiles

vimrc

```
" Abbreviations {{{
abbr guys folks
abbr shruggie ¯\_( )_/¯
" }}}

" Keymappings {{{
"left/right arrows to switch buffers in normal mode
map <right> :bn<cr>
map <left> :bp<cr>
map <home> :rewind<cr>
map <end> :last<cr>
map g<Tab> :bn<CR>
" Make Y behave like C and D
nnoremap Y y$
" Use , and space in addition to \ for the leader
let mapleader = ","
nmap \ ,
nmap <space> ,
" save my pinky
nore ; :
" auto-format the current paragraph
nnoremap == :call WrapMerge()<CR>
" Get rid of jumping behavior when using these search functions
```

```
nnoremap * *<c-o>
nnoremap # #<c-o>
" Clear search pattern with \\
map <silent> <Leader>\ :noh<CR>
" fix spelling of last misspelled word
nmap <F1> [s1z=<C-o>
imap <F1> <Esc>[s1z=<C-o>a
nmap <Leader>fs [s1z=<C-o>
" Poor man's cscope - grep for symbol under cursor
nnoremap gr :grep '\b<cword>\b' *<CR>
" Clean up left side
nmap <F2> :set nonu foldcolumn=0<CR>:QuickfixsignsToggle<CR>
" Show netrw sidebar
map <silent> <F9> :call ToggleVExplorer()<CR>
nnoremap <silent> <F10> :TagbarToggle<CR>
set pastetoggle=<F11>
" jump to next quickfix item
map <F12> :cn<CR>
" preview the tag under the cursor
nmap <C-p> :exe "ptag" expand("<cword>")<CR>
" Toggle the quickfix window
nnoremap <silent> <C-c> :call ToggleQuickfix()<cr>
" Window movement
nnoremap <C-j> <C-w>j
nnoremap <C-k> <C-w>k
nnoremap <C-h> <C-w>h
nnoremap <C-l> <C-l>k
" Keep selected blocks selected when shifting
vmap > >gv
vmap < <gv
" Move visual blocks up and down
vnoremap J :m '>+1<CR>gv=gv
vnoremap K :m '<-2<CR>gv=gv
" day+time / day+date
nmap <Leader>dt "=strftime("%c")<CR>P"
nmap <Leader>dd "=strftime("%y-%m-%d")<CR>P"
" Change to the directory of the current file
nmap cd :lcd %:h \| :pwd<CR>
" Fix whitespace
nmap <Leader>fw :StripWhitespace<CR>
" Quick exits
nmap zz ZZ
" }}}

" Settings {{{
syntax on
filetype plugin on
filetype indent on
```

```
"helptags ~/.vim/doc

if $DISPLAY != ""
    "set cursorline          " I like this, but damn is it slow
    set clipboard=unnamed
    set mouse=a             " Turn this off for console-only mode
    if !has('nvim')
        set ttymouse=xterm2
    endif
endif
set et                      " expand tabs
set diffopt+=iwhite,vertical,filler   " ignore whitespace in diffs
set hidden                  " allow hidden buffers
set noerrorbells vb t_vb=   " no bells
set nonu                    " line numbers
set viewdir=$HOME/.views    " keep view states out of my .vim
set pumheight=15            " trim down the completion popup menu
set shortmess+=atIoT        " save space in status messages
set scrolloff=3             " 3 lines of buffer before scrolling
set ignorecase              " case insensitive searches
set smartcase               " unless you type uppercase explicitly
set smarttab                " use shiftwidth instead of tab stops
set wildmode=longest,list   " shows a list of candidates when tab-completing
set wildmenu                " use a more functional completion menu when tab-completing
set encoding=utf-8           " always use utf-8
set hlsearch                 " highlight all search matches
set foldcolumn=0             " I never use this.
set nojoinspaces             " disallow two spaces after a period when joining
set formatoptions=qjnrtlmnc " auto-formatting style
set autoindent
set shiftround               " Round to the nearest shiftwidth when shifting
set linebreak                " When soft-wrapping long lines, break at a word
set comments-=s1:/*,mb:*,ex:*/
set comments+=fb:*,b:\\item
set formatlistpat=^\\s*\\([0-9]\\+\\\|[a-z]\\)[\\].:)}]\\s\\+
set grepprg=grep\ -R\ --exclude=\"*.aux\"\ --exclude=\"tags\"\ --exclude=\"*scope.out\"\ --color=always\ -nIH\
set cpoptions=BFt
set completeopt=menuone,longest
autocmd Filetype *
        \    if &omnifunc == "" |
        \        setlocal omnifunc=syntaxcomplete#Complete |
        \    endif
set tags=tags,./tags
set nobackup                 " ugh, stop making useless crap
set nowritebackup            " same with overwriting
set directory=/tmp           " litter up /tmp, not the CWD
set nomodeline               " modelines are dumb
set tabstop=4 shiftwidth=4 softtabstop=4
set backspace=indent,eol,start
set ruler                   " show position in file
set title
set titlestring=%t%(\ %M%)%(\ (%{expand(\"%:p:h\")})%)%(\ %a%)
set titleold=""
set ttimeout
set ttimeoutlen=100         " Make it so Esc enters Normal mode right away
if has('nvim')
    set ttimeoutlen=-1
endif
set helpheight=0            " no minimum helpheight
set incsearch               " search incrementally
set showmatch               " show the matching terminating bracket
set suffixes=.out           " set priority for tab completion
set sidescroll=1            " soft wrap long lines
set lazyredraw ttyfast      " go fast
set errorfile=/tmp/errors.vim
set cscopequickfix=s-,c-,d-,i-,t-,e-        " omfg so much nicer
set foldlevelstart=0        " the default level of fold nesting on startup
set autoread                " Disable warning about file change to writable
set conceallevel=0          " Don't hide things by default
set wildignore+=*.bak,~*,*.o,*.aux,*.dvi,*.bbl,*.blg,*.orig,*.toc,*.fls
set wildignore+=*.loc,*.gz,*.tv,*.ilg,*.lltr,*.lov,*.lstr,*.idx,*.pdf
set wildignore+=*.fdb_latexmk,*.ind,*.cg,*.tdo,*.log,*.latexmain,*.out
```

```
" Use pipe cursor on insert
let &t_SI = "\<esc>[5 q"
let &t_SR = "\<esc>[3 q"
let &t_EI = "\<esc>[1 q"
" Same in neovim
let $NVIM_TUI_ENABLE_CURSOR_SHAPE = 1

" colors
if $TERM == 'xterm-kitty'
    if exists('$TMUX')
        set t_Co=256
    else
        set termguicolors
    endif
else
    set t_Co=256
endif
colorscheme lx-truecolor
" }}}

" Plugins
call plug#begin('~/.vim/plugged')
Plug 'AndrewRadev/id3.vim'
Plug 'ConradIrwin/vim-bracketed-paste'
Plug 'Raimondi/delimitMate'
Plug 'SirVer/ultisnips' | Plug 'honza/vim-snippets'
Plug 'SolaWing/vim-objc-syntax'
Plug 'Yggdroot/indentLine', { 'on': 'IndentLinesEnable' }
Plug 'ajh17/VimCompletesMe'
Plug 'blindFS/vim-taskwarrior'
Plug 'brookhong/cscope.vim'
Plug 'christianrondeau/vim-base64'
Plug 'd0c-s4vage/pct-vim'
Plug 'fidian/hexmode'
Plug 'godlygeek/tabular'
Plug 'goldfeld/vim-seek'
Plug 'gorkunov/smartpairs.vim'
Plug 'guns/xterm-color-table.vim'
Plug 'hhvm/vim-hack'
Plug 'jamessan/vim-gnupg'
Plug 'jceb/vim-editqf'
Plug 'jremmen/vim-ripgrep'
Plug 'junegunn/fzf', { 'dir': '~/.fzf', 'do': './install --all' }
Plug 'junegunn/fzf.vim'
Plug 'junegunn/gv.vim'
Plug 'junegunn/vim-fnr'
Plug 'junegunn/vim-pseudocl'
```

```vim
Plug 'junegunn/vim-slash'
Plug 'kergoth/vim-hilinks'
Plug 'lervag/vimtex', { 'for': 'latex' }
Plug 'majutsushi/tagbar', { 'on': 'TagbarToggle'}
Plug 'millermedeiros/vim-statline'
Plug 'ntpeters/vim-better-whitespace'
Plug 'rhysd/clever-f.vim'
Plug 'solarnz/thrift.vim', { 'for': 'thrift'}
Plug 'tomtom/quickfixsigns_vim'
Plug 'tpope/vim-afterimage'
Plug 'tpope/vim-characterize'
Plug 'tpope/vim-eunuch'
Plug 'tpope/vim-fugitive'
Plug 'tpope/vim-obsession'
Plug 'tpope/vim-repeat'
Plug 'tpope/vim-surround'
Plug 'tpope/vim-vinegar'
Plug 'vim-utils/vim-man'
Plug 'will133/vim-dirdiff'
call plug#end()

" Don't load plugins that have unmet dependencies
if !executable('task')
    let g:loaded_taskwarrior = 1
```

```
endif

if !has('python3')
    let g:loaded_pct = 1
endif

" netrw {{{
let g:netrw_liststyle=0
let g:netrw_browse_split=4
let g:netrw_winsize=25
let g:netrw_banner=0
"let g:netrw_list_hide='\(^\|\s\s\)\zs\.\S\+' "hide files by default
"let g:netrw_sort_sequence = '[\/]$,*,\%(' . join(map(split(&suffixes, ','), 'escape(v:val, ".*$~")'), '\|') .
" }}}

" quickfixsigns {{{
let g:quickfixsigns_classes=['qfl', 'loc', 'marks', 'vcsdiff', 'breakpoints']
let g:quickfixsigns_echo_balloon = 1
" Disable display of the ' and . marks, so the gutter will be disabled until
" manually set marks or quickfix/diff info is present.
let g:quickfixsigns#marks#buffer = split('abcdefghijklmnopqrstuvwxyz', '\zs')
let g:quickfixsign_use_dummy = 0
let g:quickfixsigns#vcsdiff#highlight = {'DEL': 'QuickFixSignsDiffDeleteLx', 'ADD': 'QuickFixSignsDiffAddLx', '
" }}}
" buftabline {{{
let g:buftabline_show=1
let g:buftabline_separators=1
" }}}

" clever-f {{{
let g:clever_f_mark_char_color="PreProc"
let g:clever_f_smart_case=1
" }}}

" ultisnips {{{
let g:UltiSnipsExpandTrigger = "<C-l>"
" }}}

" cscope {{{
let g:cscope_interested_files = '\.java$\|\.php$\|\.h$\|\.hpp|\.cpp|\.c$|\.m$|\.swift$|\.py$|\.hs$'
let g:cscope_split_threshold = 99999
let g:cscope_auto_update = 0
nnoremap <leader>fa :call CscopeFindInteractive(expand('<cword>'))<CR>
nnoremap <leader>l :call ToggleLocationList()<CR>
nnoremap  <leader>fs :call CscopeFind('s', expand('<cword>'))<CR>
nnoremap  <leader>fg :call CscopeFind('g', expand('<cword>'))<CR>
nnoremap  <leader>fd :call CscopeFind('d', expand('<cword>'))<CR>
nnoremap  <leader>fc :call CscopeFind('c', expand('<cword>'))<CR>
nnoremap  <leader>ft :call CscopeFind('t', expand('<cword>'))<CR>
nnoremap  <leader>fe :call CscopeFind('e', expand('<cword>'))<CR>
nnoremap  <leader>ff :call CscopeFind('f', expand('<cword>'))<CR>
nnoremap  <leader>fi :call CscopeFind('i', expand('<cword>'))<CR>
" }}}

" Indentlines {{{
nmap \|\| :IndentLinesToggle<CR>
let g:indentLine_faster = 1
let g:indentLine_enabled = 0
" }}}

" ripgrep {{{
let g:rg_highlight = 1
" "}}}

" FZF {{{
set rtp+=~/.fzf
set rtp+=/usr/local/opt/fzf
nmap <C-e> :Files<CR>
nmap <C-g> :GFiles<CR>
nmap <leader>m :History<CR>
nmap <leader>e :Files<CR>
nmap <Leader>t :Tags<CR>
nmap <Leader>b :BTags<CR>
nmap <C-]> :call fzf#vim#tags(expand('<cword>'), {'options': '--exact --select-1 --exit-0'})<CR>

let g:fzf_tags_command = '/usr/local/bin/ctags -R'

function! s:build_quickfix_list(lines)
  call setqflist(map(copy(a:lines), '{ "filename": v:val }'))
  copen
  cc
endfunction

let g:fzf_action = {
  \ 'ctrl-q': function('s:build_quickfix_list'),
  \ 'ctrl-t': 'tab split',
  \ 'ctrl-x': 'split',
  \ 'ctrl-v': 'vsplit' }

command! -bang -nargs=* F
  \ call fzf#vim#grep(
  \   'rg --column --line-number --no-heading --fixed-strings --ignore-case --no-ignore --hidden --follow --glo
  \   <bang>0 ? fzf#vim#with_preview('up:60%')
  \           : fzf#vim#with_preview('right:50%:hidden', '?'),
  \   <bang>0)
```

```
command! -bang -nargs=? -complete=dir Files
  \ call fzf#vim#files(<q-args>, fzf#vim#with_preview(), <bang>0)
" }}}

" statline {{{
let g:statline_fugitive=1
let g:statline_trailing_space=0
let g:statline_mixed_indent=0
let g:statline_filename_relative=1
let g:statline_show_encoding=0
" }}}

" tagbar {{{
let g:tagbar_iconchars = [' ', ' ']
" }}}

" augroups {{{
augroup cjava
    au!
    au BufWinEnter *.[mCchly] set nospell number comments+=s1:/*,mb:*,ex:*/
    au BufWinEnter,BufNewFile *.m,*.xm,*.xmi setfiletype objc
    au BufWinEnter,BufNewFile *.m,*.xm,*.xmi let c_no_curly_error = 1
    au BufWinEnter *.cpp,*.java,*.hs set nospell number
    au BufWinLeave *.[mchly] mkview
    au BufWinEnter *.[mchly] silent loadview
    au BufWinLeave *.cpp,*.java,*.hs mkview
    au BufWinEnter *.cpp,*.java,*.hs silent loadview
augroup end

augroup html
    au!
    au FileType html set spell wrapmargin=5 wrapscan number
    au FileType html set wrapscan&
    au BufWinLeave *.htm* mkview
    au BufWinEnter *.htm* silent loadview
augroup end

augroup pythonphp
    au FileType python set smartindent smarttab nospell number
    au FileType php set smartindent smarttab nospell number
    au BufWinLeave *.py,*.php mkview
    au BufWinEnter *.py,*.php silent loadview
augroup end

augroup markdown
    " Don't highlight underscores
    syn match markdownError "\w\@<=\w\@="
    au BufWinEnter *.notes set filetype=markdown
    au BufWinLeave *.md,*.notes, mkview
    au BufWinEnter *.md,*.notes, silent loadview
    au BufWinEnter *.md,*.notes, imap <C-l> <C-t>
    au BufWinEnter *.md,*.notes, imap <C-h> <C-d>
    au BufWinEnter *.md,*.notes, normal zR
    au BufWinEnter *.md,*.notes,*mutt*, imap >> <C-t>
    au BufWinEnter *.md,*.notes,*mutt*, imap << <C-d>
    au FileType markdown set spell
    au FileType markdown set textwidth=78 complete+=k comments+=b:-,b:+,b:*,b:+,n:>
augroup end

" Disable spellcheck on quickfix, switch between quickfix lists with the arrow
" keys
augroup quickfix
    au FileType qf, noremap ' <CR><C-W><C-P>j
    au FileType qf, set nospell number
    au FileType qf, nnoremap <silent> <buffer> <right> :cnew<CR>
    au FileType qf, nnoremap <silent> <buffer> <left> :col<CR>
    au FileType qf, setlocal statusline=\ %n\ \ %f%=L%l/%L\ %P
    au BufReadPost quickfix call GrepColors()
    au BufWinEnter quickfix call GrepColors()
    au BufWinEnter qf:list call GrepColors()
augroup end
augroup msdocs
    au BufReadCmd *.docx,*.xlsx,*.pptx call zip#Browse(expand("<amatch>"))
    au BufReadCmd *.odt,*.ott,*.ods,*.ots,*.odp,*.otp,*.odg,*.otg call zip#Browse(expand("<amatch>"))
augroup end

augroup misc
    au FileType git set foldlevel=99
    au BufWinEnter *.applescript set filetype=applescript
    au BufWinEnter *.nmap, set syntax=nmap
    au BufWinEnter *.scala, set filetype=scala
    au BufWinEnter *.proto, set filetype=proto
    au BufWinEnter *.dtrace, set filetype=D
    au BufWinEnter *.less, set filetype=css
    au BufWinEnter *.fugitiveblame,*.diff, set nospell number
    au BufWinLeave *.txt,*.conf,.vimrc,*.notes mkview
    au BufWinEnter *.txt,*.conf,.vimrc,*.notes silent loadview
    au BufWinEnter .vimrc set foldmethod=marker
    au FileType json set conceallevel=0
    au FileType make set diffopt-=iwhite
    au FileType vim set nospell
    au FileType mail set spell complete+=k nonu
    au FileType mail if executable("par") | set formatprg=par | endif
    au FileType mail map <F8> :%g/^> >/d<CR>gg10j
    au FileType mail StripWhitespace
    au FileType mail,text let b:delimitMate_autoclose = 0
    au BufWinEnter *vimChatRoster, set foldlevel=1
    au BufWinEnter *.nse set filetype=lua
    " If a JS file has only one line, unminify it
    au FileType javascript if line('$')==1 | call Unminify() | endif
    au FileType help set nospell
    " What - like how does this even work
    au InsertLeave * hi! link CursorLine CursorLine
    au InsertEnter * hi! link CursorLine Normal
    " Disable the 'warning, editing a read-only file' thing that
    " hangs the UI
    au FileChangedRO * se noreadonly
    au GUIEnter * set visualbell t_vb=
augroup end

" }}}

" Custom functions {{{
" Quickfix toggle

let g:quickfix_is_open = 0

function! ToggleQuickfix()
    if g:quickfix_is_open
```

```
        cclose
        let g:quickfix_is_open = 0
        execute g:quickfix_return_to_window . "wincmd w"
    else
        let g:quickfix_return_to_window = winnr()
        copen
        let g:quickfix_is_open = 1
    endif
endfunction

" Toggle Vexplore
function! ToggleVExplorer()
  if exists("t:expl_buf_num")
      let expl_win_num = bufwinnr(t:expl_buf_num)
      if expl_win_num != -1
          let cur_win_nr = winnr()
          exec expl_win_num . 'wincmd w'
          close
          exec cur_win_nr . 'wincmd w'
          unlet t:expl_buf_num
      else
          unlet t:expl_buf_num
      endif
  else
```

```
      exec '1wincmd w'
      Vexplore
      let t:expl_buf_num = bufnr("%")
  endif
endfunction

" wrap nicely
function! WrapMerge()
    set formatoptions-=w
    exec "normal gwip"
    set formatoptions+=w
endfunction

" Read in cookiefiles
command -bar Cookies call ReadCookies()
function ReadCookies()
    call system("cp Cookies.binarycookies /tmp/")
    %!python $HOME/bin/BinaryCookieReader.py /tmp/Cookies.binarycookies
endfunction

" I use this to highlight the match from grep, but keep quickfix syntax
" highlighting intact. Detects Linux due to the different escape sequences of
" GNU grep.
command -bar GrepColors call GrepColors()
```

```
function GrepColors()
    set conceallevel=3
    set cocu=nv

    if system('uname')=~'Linux'
        syn region ansiRed start="\e\[01;31m"me=e-2 end="\e\[m"me=e-3 contains=ansiConceal
        syn match ansiConceal contained conceal    "\e\[\(\d*;\)*\d*m"
        syn match ansiStop        conceal "\e\[m"
   elseif system('uname')=~'FreeBSD'
       syn region ansiRed start="\e\[01;31m"me=e-2 end="\e\[00m"me=e-5 contains=ansiConceal
       syn match ansiConceal contained conceal    "\e\[\(\d*;\)*\d*m"
       syn match ansiStop        conceal "\e\[00m\e\[K"
    else
        syn region ansiRed start="\e\[01;31m\e\[K"me=e-2 end="\e\[m"me=e-3 contains=ansiConceal
        syn match ansiConceal contained conceal    "\e\[\(\d*;\)*\d*m\e\[K"
        syn match ansiStop        conceal "\e\[m\e\[K"
    endif

    hi ansiRed    ctermfg=197   guifg=#FF005F  cterm=none         gui=none
    hi! link ansiStop NONE
endfunction

" Simple re-format for minified Javascript
command! Unminify call Unminify()
function! Unminify()
    %s/{\ze[^\r\n]/{\r/g
    %s/){/) {/g
    %s/};\?\ze[^\r\n]/\0\r/g
    %s/;\ze[^\r\n]/;\r/g
    %s/[^\s]\zs[=&|]\+\ze[^\s]/ \0 /g
    normal ggVG=
endfunction

command! -nargs=1 Graudit call Graudit(<f-args>)
function! Graudit(db)
    call system("$HOME/git/graudit/graudit -B -x 'cscope.*' -c0 -d " . a:db . " . | awk 'length($0) < 200' > /t
    copen
    cf /tmp/graudit.out
endfunction
" }}}
```