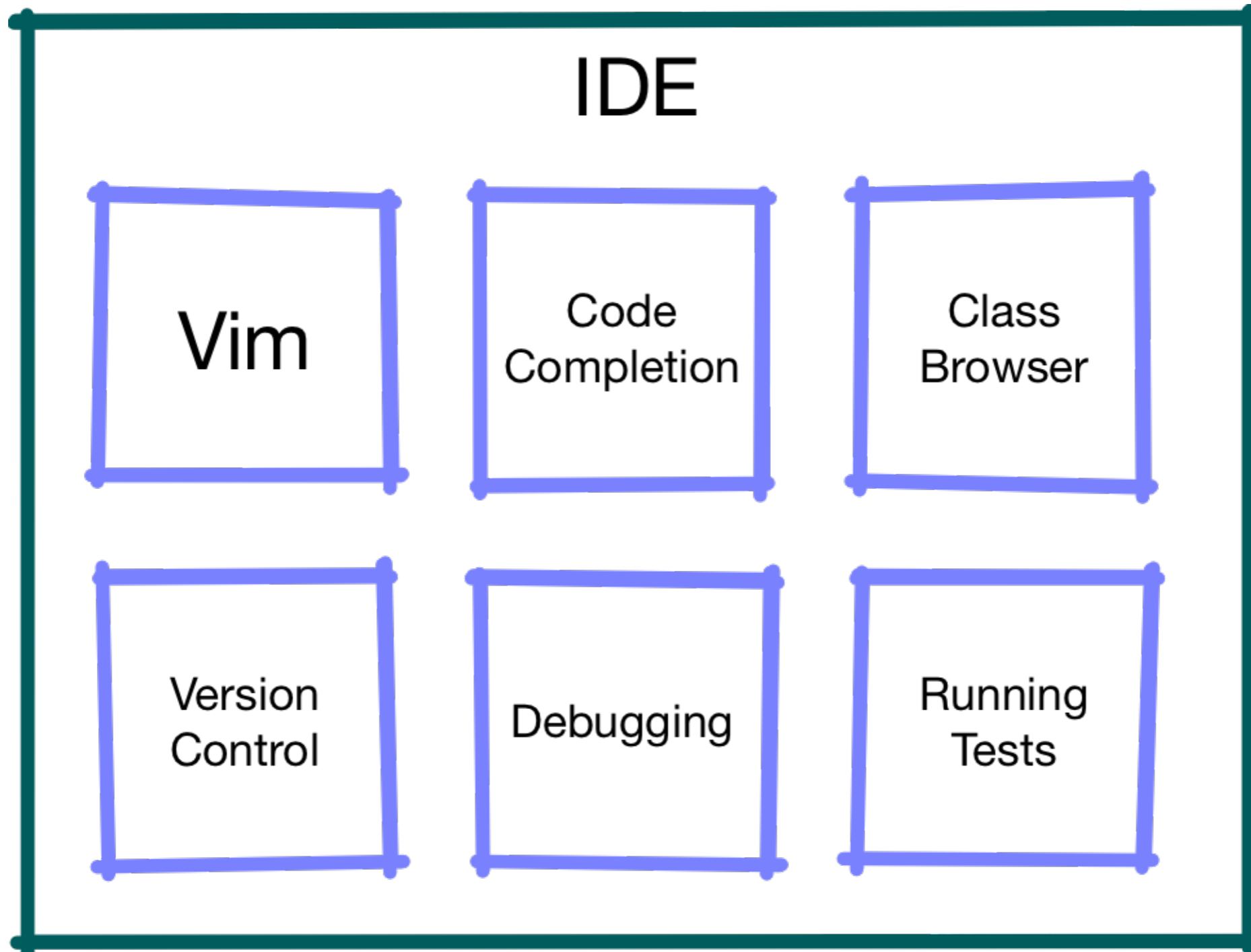"Vim the IDE"

about.txt

— US Guy in Hamburg

— Software Engineer @ Yelp in Hamburg

disclaimer: opinions expressed are solely my own and do not express the views or opinions of my employer.

"Vim the Editor"

Vim

The IDE has many parts

Problems with the IDE

— too many features
— much complexity
— same with "All in 1 Vim Builds"

IDE Goals

Navigation

Finding things
- in the current file
- other files
- searching via regex, symbols, or simple terms

You clone a new project and either
- know where you are going
- want to find someplace to go

Finding Things

— ctrlp https://github.com/ctrlpvim/ctrlp.vim

— py-matcher to increase speed https://github.com/FelikZ/ctrlp-py-matcher

The main fuzzy (and regex compatible) way to jump between files.
Nearly instant with 60K files (linux source).

- go jump to file `map gj`
- buffer switching `map gb`
- recent files `map gr`

Project Navigation

— NERDTree https://github.com/scrooloose/nerdtree

— NERDTree Tabs https://github.com/jistr/vim-nerdtree-tabs

— vim-vinegar https://github.com/dhruvasagar/vim-vinegar

Always quickly available with '-' key. Easy to switch to files in the same directory or see where you are.

```
README    ~/D/linux-4.13.3 |
 1 Linux kernel
 2 ============
 3
 4 This file was moved to Documentation/admin-guide/README.rst
 5
 6 Please notice that there are several guides for kernel develope
 7 These guides can be rendered in a number of formats, like HTML
 8
 9 In order to build the documentation, use ``make htmldocs`` or
10 ``make pdfdocs``.
11
12 There are various text files in the Documentation/ subdirectory,
13 several of them using the Restructured Text markup notation.
14 See Documentation/00-INDEX for a list of what is contained in e
15
16 Please read the Documentation/process/changes.rst file, as it c
17 requirements for building and running the kernel, and informati
18 the problems which may result by upgrading your kernel.
```

```
 1 /Users/mbehrens/Downloads/linux-4.13.3/
 2 ▶ .git/
 3 ▶ arch/
 4 ▶ block/
 5 ▶ certs/
 6 ▶ crypto/
 7 ▶ Documentation/
 8 ▶ drivers/
 9 ▶ firmware/
10 ▶ fs/
11 ▶ include/
12 ▶ init/
13 ▶ ipc/
14 ▶ kernel/
15 ▶ lib/
16 ▶ mm/
17 ▶ net/
18 ▶ samples/
19 ▶ scripts/
20 ▶ security/
21 ▶ sound/
22 ▶ tools/
23 ▶ usr/
24 ▶ virt/
25   .cocciconfig
26   .get_maintainer.ignore
27   .gitattributes
28   .gitignore
29   .mailmap
30   COPYING
31   CREDITS
32   Kbuild
33   Kconfig
34   MAINTAINERS
35   Makefile
36   README
37   tags
```

```
 1 Linux kernel
 2 ============
 3
 4 This file was moved to Documentation/admin-guide/README.rst
 5
 6 Please notice that there are several guides for kernel develop
 7 These guides can be rendered in a number of formats, like HTML
 8
 9 In order to build the documentation, use ``make htmldocs`` or
10 ``make pdfdocs``.
11
12 There are various text files in the Documentation/ subdirector
13 several of them using the Restructured Text markup notation.
14 See Documentation/00-INDEX for a list of what is contained in
15
16 Please read the Documentation/process/changes.rst file, as it
17 requirements for building and running the kernel, and informat
18 the problems which may result by upgrading your kernel.
```

```
                              ~
                                        [No Name]
NERDTree Menu. Use j/k/enter and the shortcuts indicated
================================================================
> (a)dd a childnode
  (m)ove the current node
  (d)elete the current node
  (r)eveal in Finder the current node
  (o)pen the current node with system editor
  (q)uicklook the current node
  (c)opy the current node
  (l)ist the current node
```

— easy to modify files inline

— files are easier to modify with a "GUI"

Searching for things

— vim-grepper https://github.com/mhinz/vim-grepper

```
:Grepper -tool rg -query "foo"
```

2 main ways to search for things:
- quickfix search
- side search, with +3 lines context

Easy visual search
```
nmap gv <plug>(GrepperOperator)
```

Jumping to symbols

vim-gutentags makes `tags` work transparently

Editing

Mostly normal VIM things
- syntax highlighting
- linting
- code completion
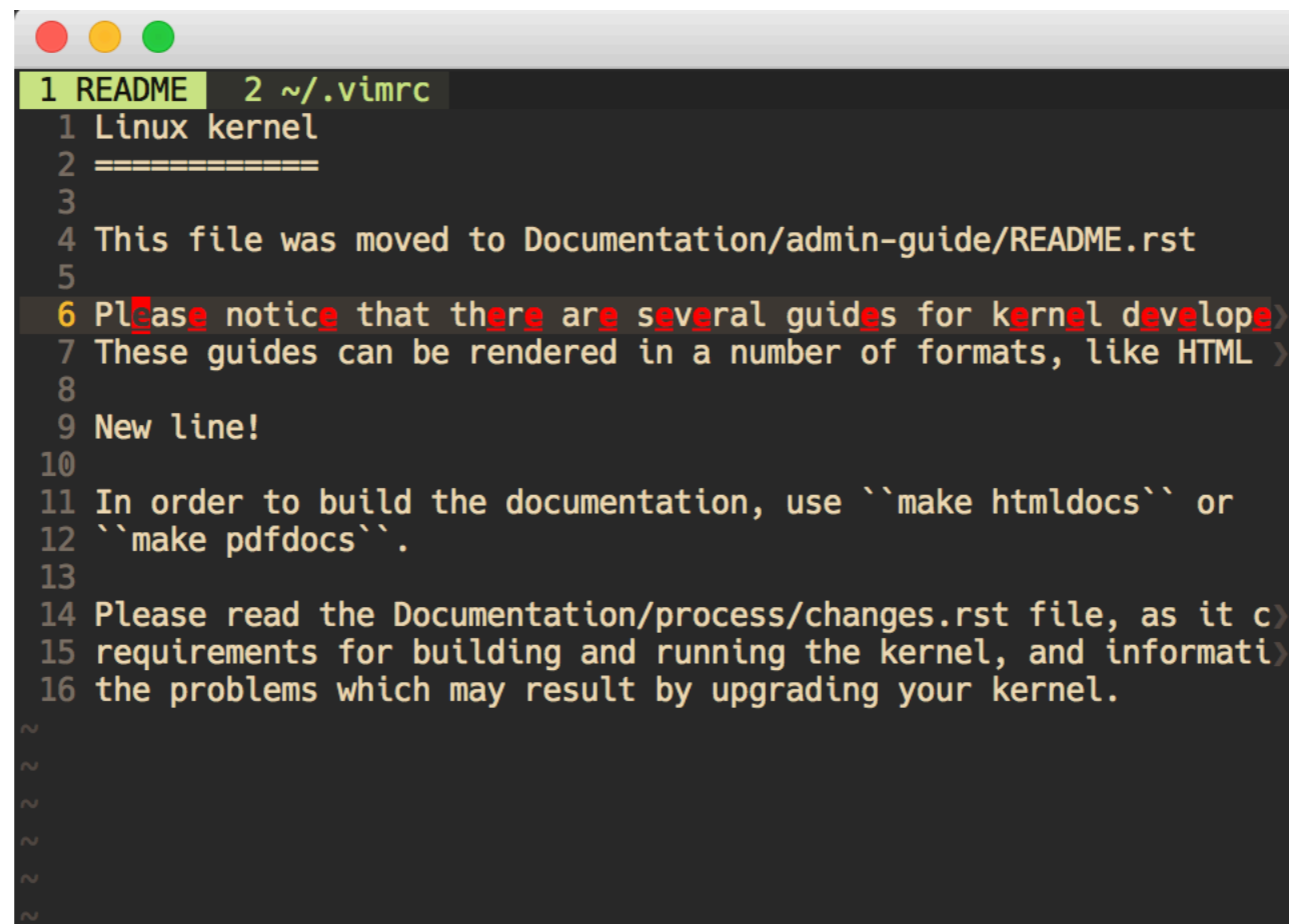
Tab Complete

— <u>SuperTab</u> configure Tab complete

— <u>vim-snipmate</u> makes snippets work

— <u>vim-snippets</u> default snippets for everything

Easier jumping to characters

— vim-clever-f
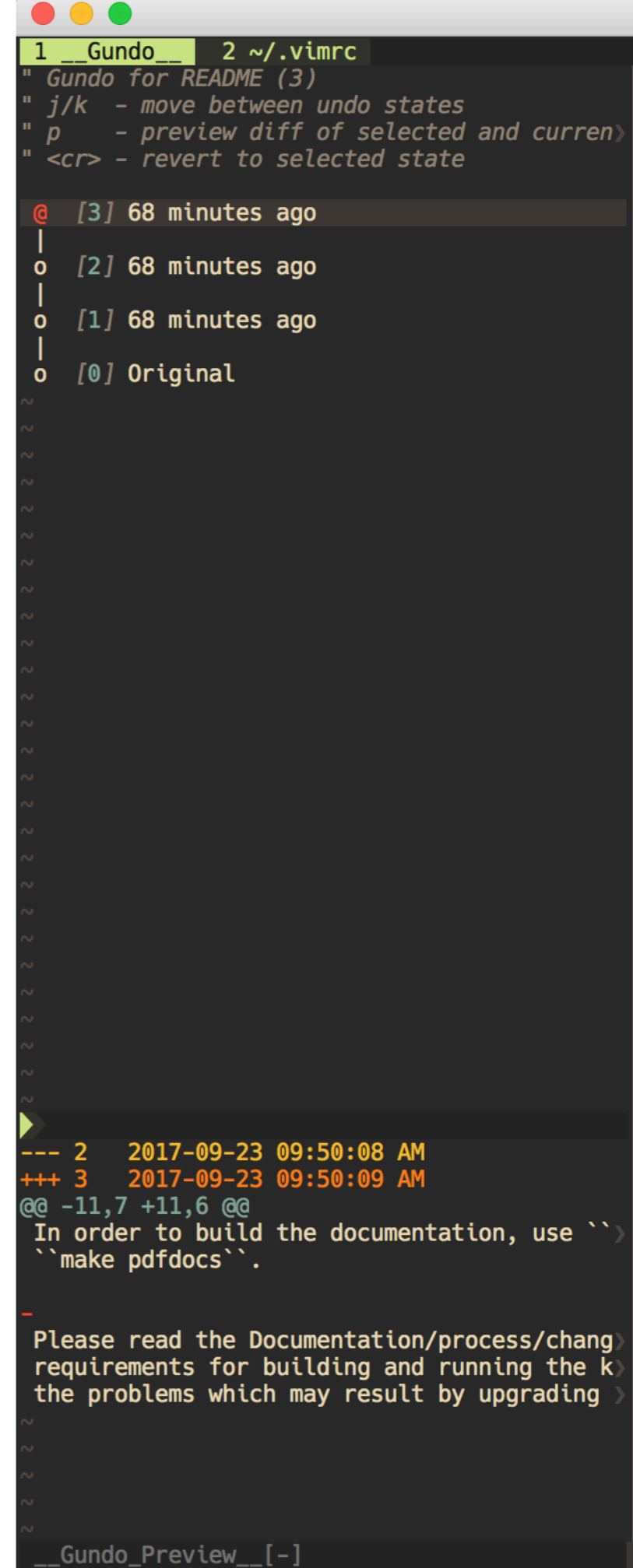
Find the next character: type `fe`

Easy revisions

— <u>vim-gundo</u>

Easy alignment

— <u>vim-lion</u>

Working with tag based languages

— <u>vim-surround</u>

External Interactions

— Building software (compiling, make)

— Version Control

— Running tests

— Debugging

Building Software

```
set makeprg=make
```

Testing!

— https://github.com/benmills/vimux

The future:

— https://github.com/janko-m/vim-test
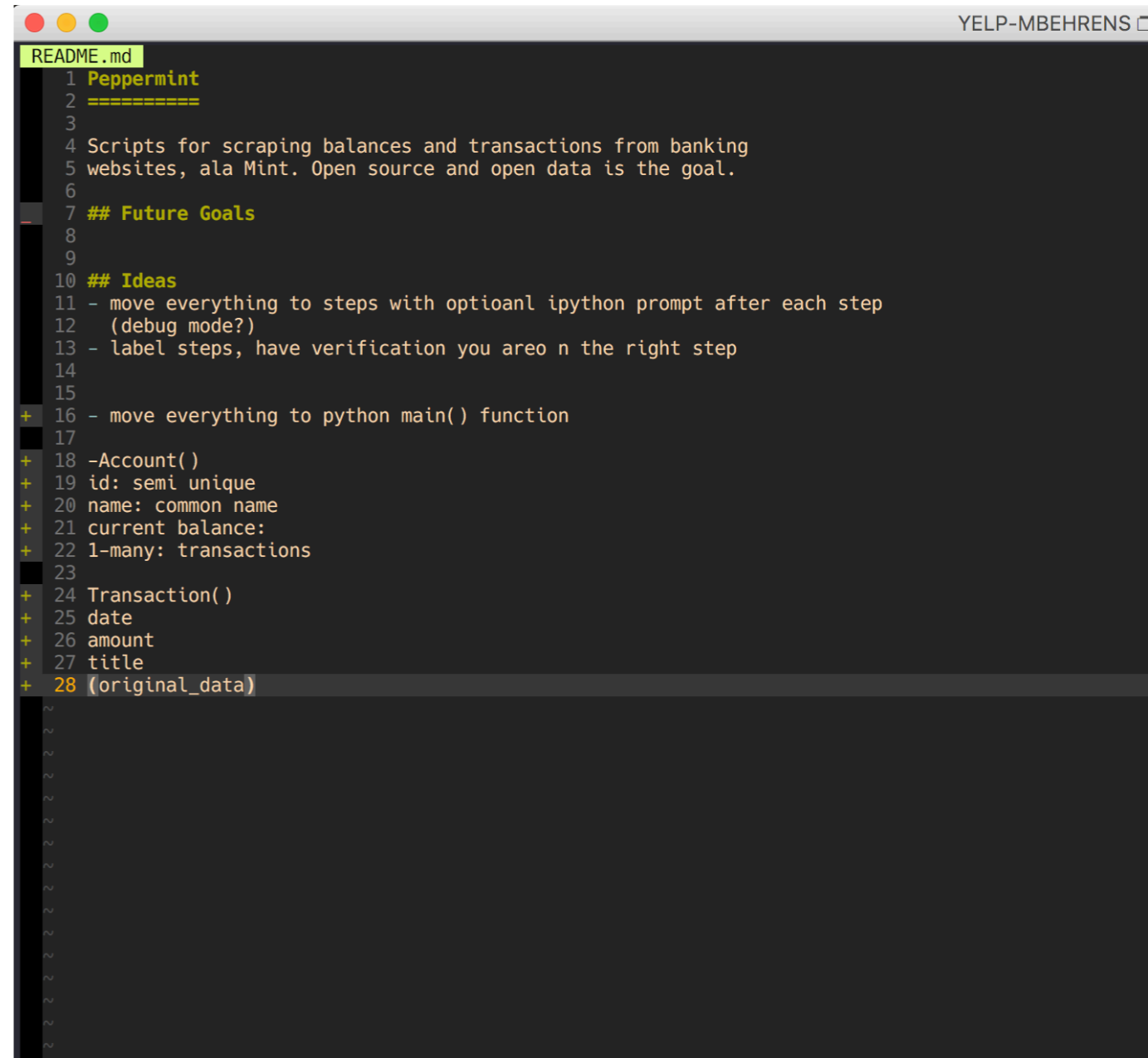
Structure your tests to mirror your code!

```
python/package/foo.py
tests/package/foo_test.py
```

Version Control

— <u>git-gutter</u>
— <u>vim-fugitive</u>

```
README.md
    1  Peppermint
    2  ==========
    3
    4  Scripts for scraping balances and transactions from banking
    5  websites, ala Mint. Open source and open data is the goal.
    6
    7  ## Future Goals
    8
    9
   10  ## Ideas
   11  - move everything to steps with optioanl ipython prompt after each step
   12    (debug mode?)
   13  - label steps, have verification you areo n the right step
   14
   15
   16  - move everything to python main() function
   17
   18  -Account()
   19  id: semi unique
   20  name: common name
   21  current balance:
   22  1-many: transactions
   23
   24  Transaction()
   25  date
   26  amount
   27  title
   28  (original_data)
```
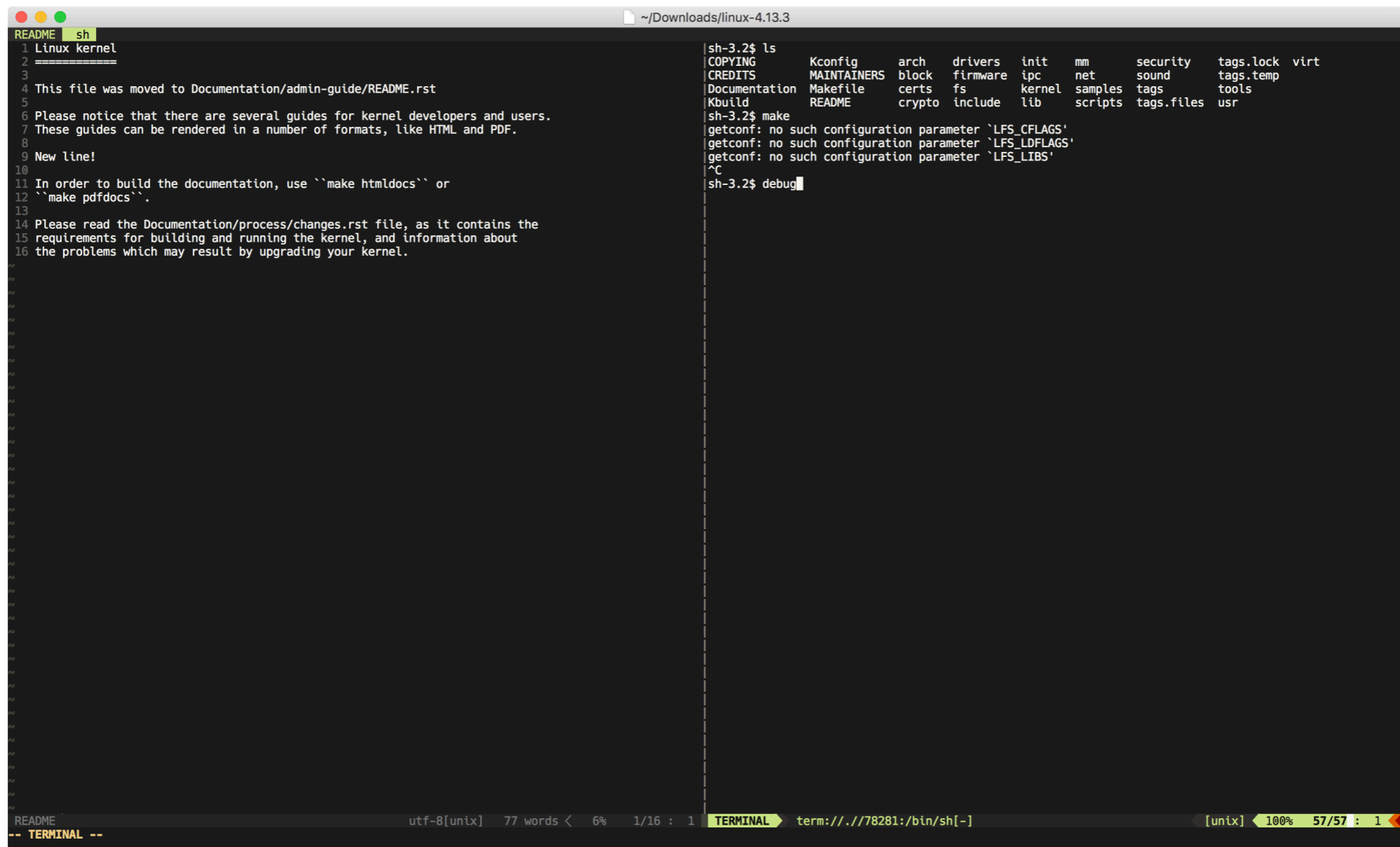
Debugging

uh oh

NeoVim

The future?

references.txt

— http://cewing.github.io/training.codefellows/assignments/day01/sublime_as_ide.html

— my dotfiles and vimrc github.com/askedrelic/dotfiles

thanks.txt

Discussion is welcome!

@askedrelic
www.asktherelic.com