SKILLPILLS

Skill Pill: Terminal

Text Editor: Vim

James Schloss

Okinawa Institute of Science and Technology
james.schloss@oist.jp

April 6, 2017

OIST

Overview

SKILLPILLS

Text Editing

Here's a basic question: **why do you want to edit text?**

Text Editing

Here's a basic question: **why do you want to edit text?**

Reason:

- Work documents
- E-mail
- Code / scripting

Appropriate editor:

- Office suite
- E-mail client
- IDE / text editor

Text Editing

Here's a basic question: **why do you want to edit text?**

Reason:

- Work documents
- E-mail
- Code / scripting

Appropriate editor:

- Office suite
- E-mail client
- IDE / text editor

Text editors are essential tools for your belt – especially when working on UNIX systems like Sango / supercomputers

Text Editing

Text editors must only edit text, **they should not inject unnecessary formatting into your file without your permission!**

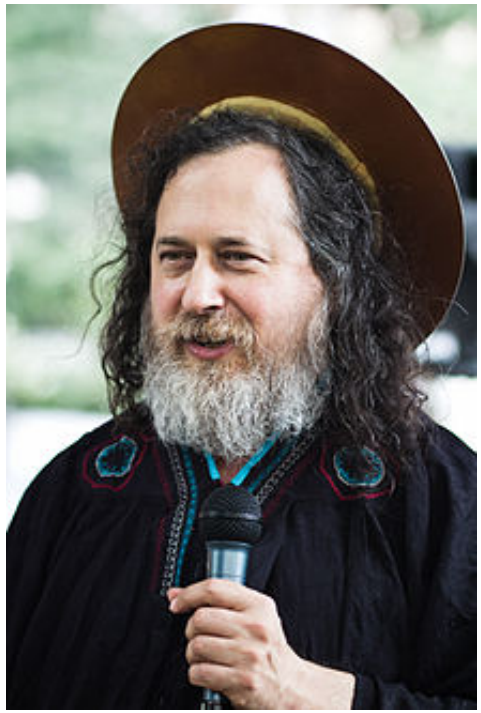
We have a few options:

- Graphical Editors
  - Sublime Text
  - Atom
  - gedit, kedit, notepad++
- Terminal Editors
  - Vi / Vim
  - Nano
  - Emacs

The Church of Emacs

Programmers are silly

- There are 2 rather famous text editors, **Vi** and **Emacs**.
- Their user-bases had a bit of a feud, which turned into the **Curch of Emacs** and the **Cult of Vi**
- By day, he is Richard Stallman, but by night, he is St. IGNU-cius $\rightarrow$
- Vi users say Emacs is "a great operating system, lacking only a decent editor"
- Emacs users say we could create a psuedorandom number generator by providing a new user access to vi for the first time and asking them to quit



Reasons to learn Vim

Reasons to learn vi / vim:

- Vi is on every UNIX computer
- It is powerful
- It is easily accessible while working remotely from another machine.

Reasons to learn Vim

SKILLPILLS

Reasons to learn vi / vim:

- Vi is on every UNIX computer
- It is powerful
- It is easily accessible while working remotely from another machine.

So Let's Get To It!

Writing stuff

For the most part, you can use vim as a standard text editor by entering insert mode, typing, and then quitting.

$$\text{i} \rightarrow \text{type} \rightarrow \text{quit [ESC :wq]}$$

Writing stuff

For the most part, you can use vim as a standard text editor by entering insert mode, typing, and then quitting.

$$\text{i} \rightarrow \text{type} \rightarrow \text{quit [ESC :wq]}$$

Undo a change with *u*
in vi, undo the undo with *u*
in vim, redo things with CTRL-R

Writing stuff

For the most part, you can use vim as a standard text editor by entering insert mode, typing, and then quitting.

$$\text{i} \rightarrow \text{type} \rightarrow \text{quit [ESC :wq]}$$

Undo a change with *u*
in vi, undo the undo with *u*
in vim, redo things with CTRL-R

Write something

Open a file and write, “Hello World!”

Writing stuff

For the most part, you can use vim as a standard text editor by entering insert mode, typing, and then quitting.

$$\text{i} \rightarrow \text{type} \rightarrow \text{quit [ESC :wq]}$$

Undo a change with *u*
in vi, undo the undo with *u*
in vim, redo things with CTRL-R

Write something

Open a file and write, “Hello World!”

Let’s talk a bit about these modes

Insert mode

Vi / Vim has $\sim$ 4 modes: Navigation (normal), Insert, Command, and Visual (we will not touch on this one).

- Insert mode
  - Enter insert mode with

i	insert the character before cursor
I	insert at the start of the line
a	insert the character after cursor
A	insert at the end of the line
o	start a new line below cursor in insert mode
O	start a new line above cursor in insert mode
R	replace text starting at the character before cursor

  - Leave insert mode with **esc**

Navigation (normal) mode

SKILLPILLS

- Navigation (normal) mode
  - Move with:

h	left
j	down
k	up
l	right
G	bottom of file
gg	start of file (vim-only)
0	start of line
$	end of line

  - Delete text with

x	delete current character
X	delete character before cursor
dd	deletes an entire line
D	deletes until the end of the line

Command mode

- Command mode:
  - Execute commands with the **colon**(**:**):

:w	writes (saves) file
:q	quits vi / vim
:wq	writes and quits **at the same time**!
:q!	quits without saving
:#	goes to line number (#)
:e	opens a new file for editing
:%s/x/y/g	replaces x for y filewide (covered later!)
:set all	shows all available settings

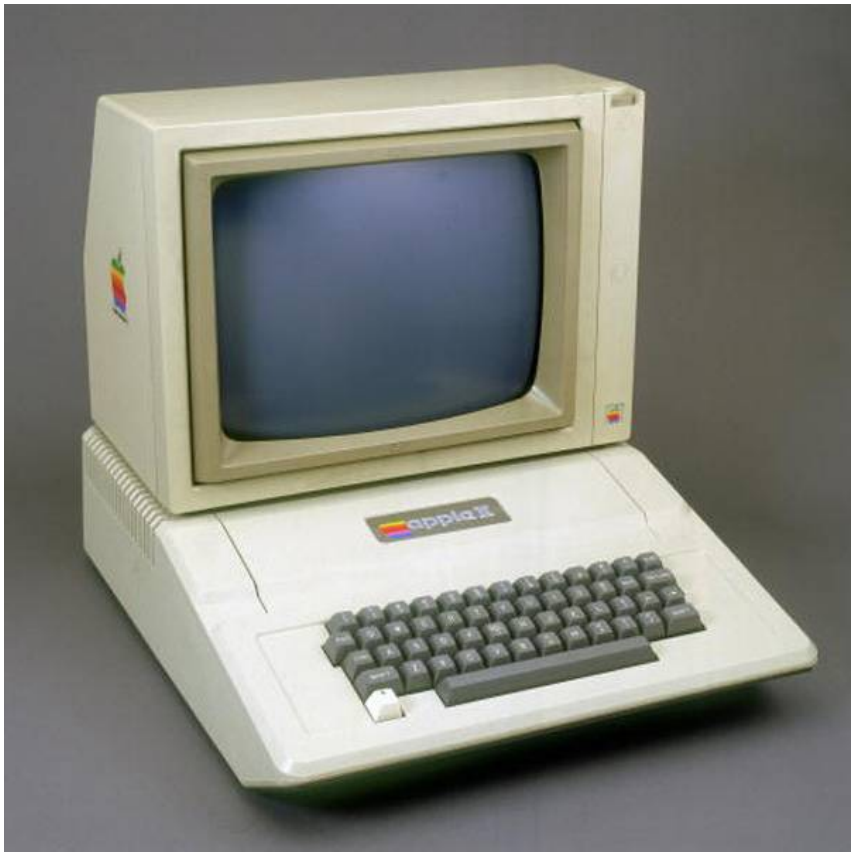
A quick historical note

For vi / vim, **which key is the most important?**

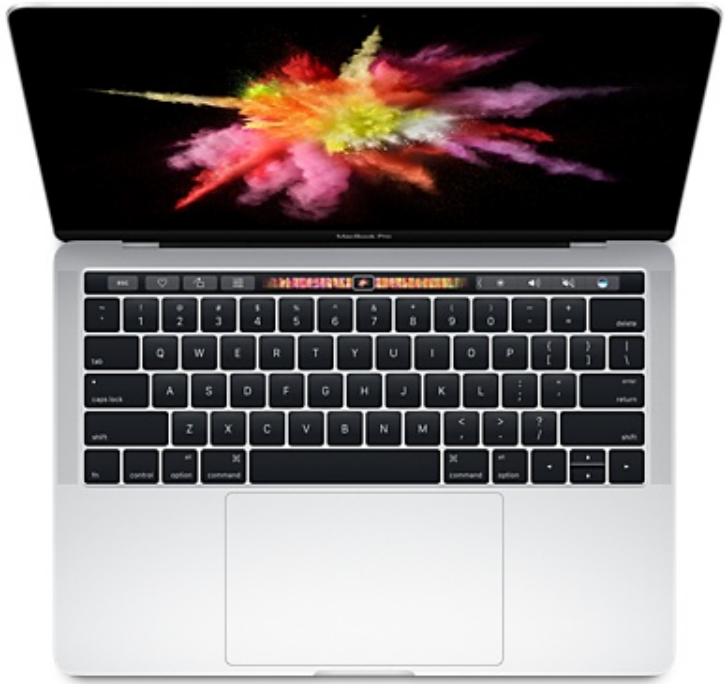
A quick historical note

For vi / vim, **which key is the most important?**

ESC



→



Where are our priorities?

Type a story into vi / vim

- Try:

```
Jack and Jill went up the hill
To fetch a pail of water.
Jack fell down and broke his crown,
And Jill came tumbling after
```

- Save the file (write and quit), and then check the file contents with **cat**

Searching

There are two ways to search in vi / vim:

/word	will bring cursor to next *word*
?word	will bring cursor to previous *word*
n/N	Proceeds with searching in forward /backwards direction
f/F c	finds character "c" in the forward / backwards direction
:set hlsearch	Sets highlighted searching in vim
:set nohlsearch	Turns off highlighted searching in vim
:noh	Turns off highlighted searching in vim until next search

Searching

There are two ways to search in vi / vim:

/word	will bring cursor to next *word*
?word	will bring cursor to previous *word*
n/N	Proceeds with searching in forward /backwards direction
f/F c	finds character "c" in the forward / backwards direction
:set hlsearch	Sets highlighted searching in vim
:set nohlsearch	Turns off highlighted searching in vim
:noh	Turns off highlighted searching in vim until next search

Find Jack

Using the story you wrote before, search for your favorite word. Become comfortable with moving in navigation (normal) mode.

The dot(.) and numbers (repeating commands)

Vi / vim have many tricks to make your life needlessly simpler

. Use this character to repeat previous command

\# Use any number before a command to repeat it # times

Put the following at the start of your story

```
################################################################################
# THIS COMMENT BLOCK IS 80 COLUMNS WIDE AND 3 COLUMNS THICK
################################################################################
```

Search and replace

The first time you need to search and replace things in vi / vim, you'll be confused.

:s/word1/word2/g Replaces *word1* with *word2* on a line

:%s/word1/word2/g Replaces *word1* with *word2* text-wide

:#1,#2s/word1/word2/g Replaces *word1* with *word2* between line numbers 1 and 2

Search and replace

The first time you need to search and replace things in vi / vim, you'll be confused.

:s/word1/word2/g Replaces *word1* with *word2* on a line

:%s/word1/word2/g Replaces *word1* with *word2* text-wide

:#1,#2s/word1/word2/g Replaces *word1* with *word2* between line numbers 1 and 2

There are four characters to use at the end:

- g This searches the space globally and will not stop at just the first occurrence every line
- c This asks for confirmation each time
- i This is case-insensitive (vim-only)
- I This is case-sensitive (vim-only)

Search and replace

The first time you need to search and replace things in vi / vim, you'll be confused.

:s/word1/word2/g Replaces *word1* with *word2* on a line

:%s/word1/word2/g Replaces *word1* with *word2* text-wide

:#1,#2s/word1/word2/g Replaces *word1* with *word2* between line numbers 1 and 2

There are four characters to use at the end:

- g This searches the space globally and will not stop at just the first occurrence every line
- c This asks for confirmation each time
- i This is case-insensitive (vim-only)
- I This is case-sensitive (vim-only)

These may be chained together:

```
:10,100s/hey/pidgeon/gci
```

Replacing Jack

SKILLPILLS

Put yourself into your story

Search for and replace a character in your story with your own name.

Yank and paste

Copy is an inferior term only used by hacker wannabe's
Yank is what real hackers say

To Copy:

yy	Yank (copy) a single line
y#y	Yank (copy) # lines
dd	Yank (copy) a single line and destroy it
d#d	Yank (copy) # lines and destroy them

To Paste

p	Pastes any yanked lines after cursor
P	Pastes any yanked lines before cursor

Yank and paste

SKILLPILLS

Copy is an inferior term only used by hacker wannabe's
Yank is what real hackers say

To Copy:

yy	Yank (copy) a single line
y#y	Yank (copy) # lines
dd	Yank (copy) a single line and destroy it
d#d	Yank (copy) # lines and destroy them

To Paste

p	Pastes any yanked lines after cursor
P	Pastes any yanked lines before cursor

BUT WAIT, THERE'S MORE!

The power of normal mode

SKILLPILLS

For copy / delete / movement, we can combine movements with actions

15j	moves down 15 lines
y5y	yanks 5 lines
d10d	deletes 10 lines
d$	deletes until the end of line
y0	yanks to the start of the line
5x	deletes 5 characters
dtc	deletes until the next “c”

Remember that these can be repeated with “.”!

A Challenge

SKILLPILLS

No Insert Mode!

Taking the story of Jack and Jill from before, write the following without using insert mode:

```
hello world
```

vimgolf

Using as few keystrokes as possible, turn this:

```
name=www-data, groups=developer
```

into this:

```
name=developer, groups=www-data
```

Vim golf

vimgolf

Using as few keystrokes as possible, turn this:

```
name=www-data, groups=developer
```

into this:

```
name=developer, groups=www-data
```

Solution:

```
fwdt,fdPldwF,PZZ
```

Settings

All possible settings can be found with *:set all*
Some possible examples are:

ai	auto indent
number	line number
compatible	turns vim into vi

These can be turned off by adding a "no" to them

noai	unsets auto indent
nonumber	unsets line numbers
nocompatible	unsets vi mode

Colorschemes

Colors are **vim-only** and require a filename extension to properly color.

:colorscheme solarized set colorscheme to *solarized*

:colorscheme ALT+TAB List all colorschemes to choose from

:syntax on/off turns coloring on and off

:NoMatchParen / :DoMatchParen sets parentheses matching

Play around a bit with options

Once you've found your optimal settings, open up your ~/.vimrc file and put the options you want in there.

Conclusions

- vi / vim is a powerful text editor that can be used to do just about everything.
- It's greatest strength is that it is found on almost every single unix machine.
- It takes time to learn and master, but it's essential to many programmer's workflow and is a great way to edit scripts and stuff quickly and efficiently.