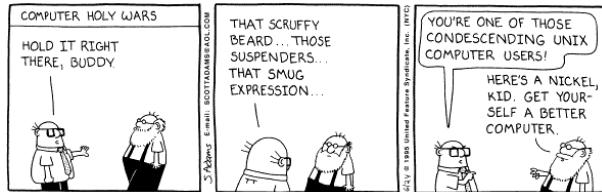


# Working with Text Files

- Objectives

- master vi or die trying



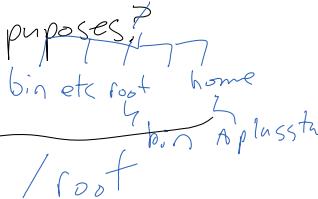
## Review

- Linus Torvalds
- Ken Thompson & Dennis Ritchie
- Richard Stallman

- What is a shell?

- What is a filesystem? What is the "top" of the filesystem?

- Name some of the standard UNIX filesystem directories and their purposes?



Text editor

- Why still use text?

- well it's usually faster...

- This is how UNIX was designed!

- No choice but to inherit this workflow...

- Unless you are Tony Stark!





- To become a system admin you need to master using a text-editor.

- Using your knowledge of Pathing the texteditor

~~will let you do any job you require.~~

- Vim or Vi (Same application practically)

- others (Graphical)

gedit

leafpad

- others (non-graphical)

nano

jed

joe

kate

nedit

emacs\*

Stream

editor

Screen oriented  
editors

- vi is more than a tool, it is a philosophy deeply rooted in the creation of UNIX. To understand vi and its purpose is to understand UNIX.

Disclaimer

- vi will appear highly strange and frustrating at first - but in learning it you will see its beauty.

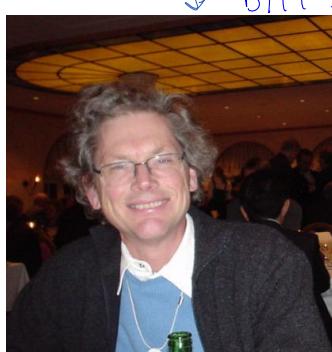
- vi history

Bill Joy

who founded



which was then purchased by Oracle





- First editor was called "ed" written by Ken Thompson for use on Teletypes (not even screens!) It worked for Thompson, but everyone else found it difficult to work with. (ed still exists!)
- 1976 "em" was created "ed for mortals"
  - increased feature set
  - But single line at a time editor



<http://linuxclues.blogspot.com/2012/09/ed-tutorial-line-editor-unix.html>

- 1977 Bill Joy at Berkley (responsible for) got a hold of it and improved "em" (BSD) called it "ex" "Extended em" "ex" went from line editor to full screen editor. Note the year... You weren't born yet...
- 1978 BSD Unix with "ex" 1.1 was released
- 1979 (I was born) most people began using ex in "visual mode" hence vi (visual ex)
- That's it... vi development basically stops...
  - Since off was part of UNIX it was illegal to use outside AT&T license.
  - In the 90's clones started to appear
  - vi is only editor standard on most UNIX/LINUX systems. Vim + Elvim are two popular alias Add color highlighting
  - Let's frame alias
  - Remember design principle is to keep fingers always on the key and moving.
- Type vi /tmp/test  
what do you think?

- The blinking you see? *(you see)*
- The Tildas ~ are your cursor ✓
- = The ~ are place holders - meaning no data yet.

No hints, no menus, no nothing - User hostile...

- Need to understand vi has two modes:

1. Command Mode

2. Input Mode

↳ used to modify / input the data

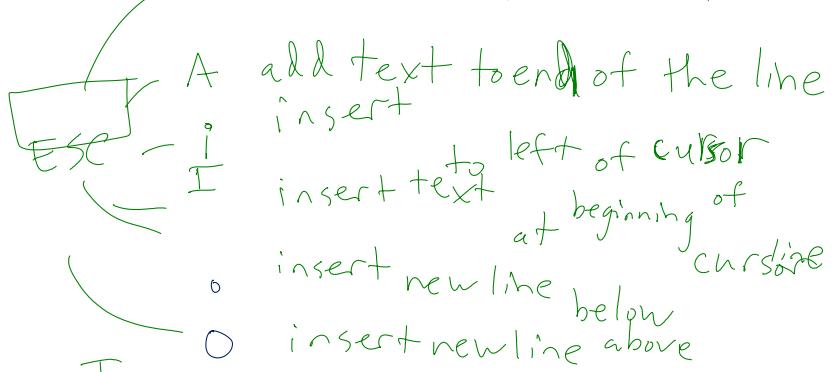
- vi starts this used to tell vi something about itself.

**[ESC]** is the toggle

Type **[Esc]** **i** to enter input mode.

Type hello world

- Some good to know commands  
a add text to the right of cursor



Try it. (Note when in **INSERT** cursor MODE  
- the word **insert** appears

(Note - when in doubt just hit **[Esc]**)

Command Mode (How to move around)

**[Esc]** out of INPUT Mode

use arrow Keys to position cursor

w - Moves cursor to beginning of next word

W - same - delimited by space

b - > opposite of 'w'

↵ - (zero) moves current line.  
 \$ - moves to end of beginning of line  
 H - Moves to current location upper  
 M - " middle left  
 L - " lower left

## Deleting, copying, changing Text

- These are more functional commands

x - deletes character

X - deletes character before cursor

d<?> - Deletes lines dd = 1 2dd 2 lines 5dd = 5 lines

y<?> - Yanks or copies text

yy = 1 2yy 2 lines ...

P = puts or pastes copied text.

period (will repeat previous command)

## Quitting

vi

ZZ Save and quit

:w write or save changes

:wq write and quit

:wq! (pronounced 'BANG') force a save

:q! exit without saving (force)

u undo

redo

ctrl + R

ctrl + G

info

Ctrl + G

last line

The : after file

the escape enters into "ex"

**ESC**

and allows you to edit lines within vi.  
 / hello searches the for  
 You can use meta-characters file  
 ^P goodby - search backwards too  
 / the \* foot  
 ^P [pP]rint

"Ex"

**Using Esc mode**

: g / Local what does do?

file it do?  
searches globally

containing "Local" for any

**Esc**

: s / Local / line

Remote ?

grep like

: g / Local / s / Remote ?

: g / Local / s / / Remote / g ?

: g / Local / s / / Remote / gp ?

More powerful

Redhat uses **updatedb** to index all files daily  
 and then the **locate** command in place of **find**  
 - **Find locate** is granular, but locate can be quicker

try sudo .bashrc  
 locate .bashrc

**Using findfind**

start location ] what to look

example

`sudo find / -name midtermexam*`

Looks for any file named midtermexam starting from root.

You can change find to `find -name` (case sensitive) or `find -iname` (case insensitive)

why? `-name` is this?

what is this?

find by sudo? size (take that locate!)

`find /usr/share -size +10M`

`find /bigdata -size +500m -size +5G`

Find all files between 500m + 5G execute the du command on them and user

- (by user name)
  - perm. (by permissions)
  - (by creation time)
  - a min (access time)
  - ctime (by creation time)
- use -not -or and all can be combined

---

global Search + replace

~~Grep~~ grep wilddesk top /etc/services /etc contain all return lines out of desktop that /services

grep -rl peerdns /usr/share/doc  
r will recursively

down the /usr/share/bash-completion/terms  
to grepfree.

You can also  
ifaddr show | grepinet → what happens?  
config pipe results

Summary -

Being able to work with plaintext & editors is crucial.  
Since most everything you need is a text file  
grep helps you search within files.

Any questions?