

vi

Or, there is no growth in the comfort zone

**Benjamin Brewster**  
OSU

# Text Editors

- There are many text editors on UNIX
  - ed
  - ex
- vi evolved from ex
  - It was written by Bill Joy in 1976 for BSD
- Its name comes from the shortest unambiguous abbreviation of visual

# vi = vicious interface

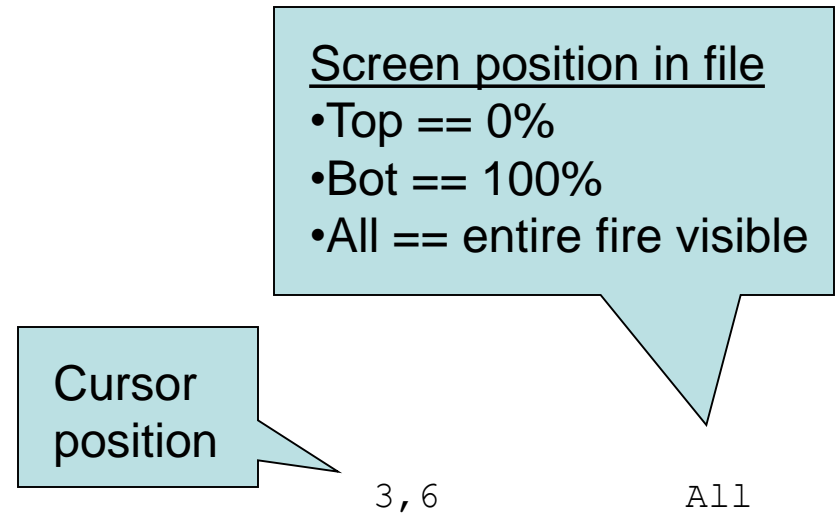
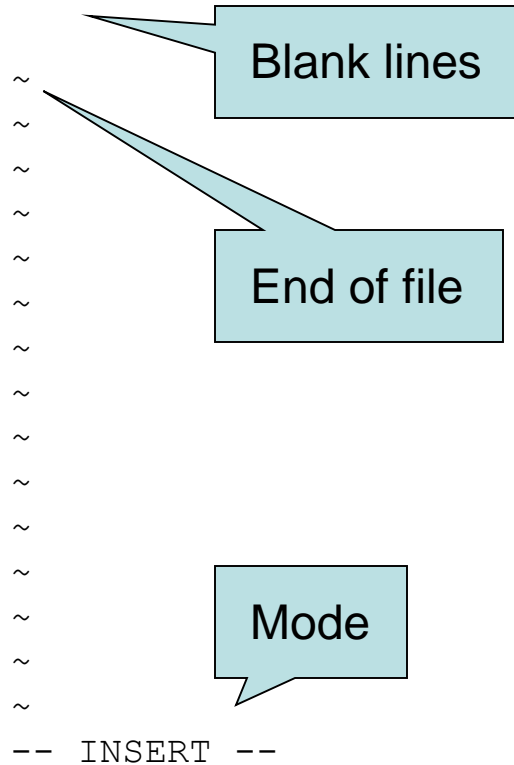
- In 1999, Tim O'Reilly, founder of the eponymous computer book publisher, stated that his company sold more copies of its vi book than its emacs book
- Not because more people like vi, I believe
- Don't try to memorize all of the keystrokes as I present them here – just be aware they exist!

# Modes, modes, modes

- vi features one of the first visual environments, instead of line editors
- Primary paradigm: vi is modal
  - Insert mode
  - Command mode
  - ...

# a view of vi(?)

I can't find the tilde key  
This is a new line  
what?



# Actually, that was vim

- vim is **vi improved**
- Better than vi in just about every way
  - Still in development
- Starting vim

```
%vim newFile  
%vim existingFile
```

# Modes for real

- Normal
  - Move around the document
  - Perform edit commands
- Insert
  - Insert text into the document
  - What normal WYSIWYG editors can only do

# Modes for real

- Replace
  - Overwrite mode
- Visual
  - Selects an area of text to which subsequent commands can be applied



# Modes for real

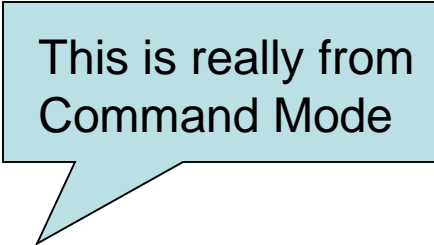
- Select
  - Kind of like selecting text on a Windows machine
  - Typing replaces everything selected
  - You're on your own, here – I have never used it
- Command
  - Whole file commands
    - Save, quit, search, etc.

# Normal Mode

- Movement

- Cursor movement

- `h, j, k, l` - the Rogue keyset
    - `$` – move to the end of the line
    - `0` – move to the beginning of the line
    - `w` – move to beginning of next word
    - `:pattern` – move to next occurrence of *pattern*
    - `n` – repeat last *pattern* search



This is really from  
Command Mode

- Screen movement

- `^d` – move screen down half a screen
    - `^u` – move screen up half a screen

- Much more later

# Normal Mode

- How to get into Normal Mode
  - vi starts in Normal Mode
  - Hit escape
    - It is always allowed to hit escape – the key can never do anything but take you to Normal mode
    - In fact, hit it a bunch of times
    - Will beep once if you're in Normal mode, and hit escape

# Insert Mode

- Type like normal



- Move around with the arrow keys
  - Commands (including movement commands) from Normal mode will not work

# Insert Mode

- How to get into insert mode
  - From Normal mode, hit `i`

i like cat`s`

In Normal Mode

`hit i`

Now in Insert Mode

i like cat`s`

`type '2'`

Still in Insert Mode

i like cat2`s`

# Insert Mode

- In this situation, use a:

i like cat

Can't move to the right!

In Normal Mode

hit a

Now in Insert Mode

i like cats

type '2'

Still in Insert Mode

i like cats2

# Replace

- Overwrite mode
  - non-insertion typing
- Two ways to get into Replace Mode
  - `r` – replace the character that the cursor is over with the next one typed
  - `R` – enter Replace Mode until we manually leave it (eg, by escape back to Normal Mode)

# Visual Mode

- Visual mode allows you to select text and then apply commands to it
- What you have selected is marked by inverted characters



# Visual Mode Demo

- Let's cut, copy, and paste:

myline

hit v

In Normal Mode

In Visual Mode

myline

hit 'l' three times

myline

hit y to yank (copy)

myline

hit p to paste

myllineine

Note the insertion point

# Command Mode

- Used to enter commands that pertain (mostly) to the entire file
- Enter Command Mode from Normal Mode by typing a colon ( : )

# Saving Files

- Enter command mode, and hit w, then enter:

`:w`

- If you're editing a new file, you'll next have to type in a name and then hit enter:

`:w myNewFileName`

# Saving Files

- “Save As” to a new file name

```
% vim newFile
```

```
...
```

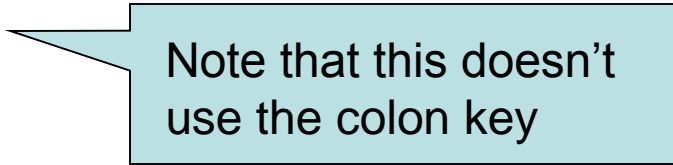
```
:w
```

```
...
```

```
:w newFile2
```

- Also, you can save to the current file and exit immediately from Normal Mode by hitting:

– ZZ



Note that this doesn't  
use the colon key

# Quitting vi

- Quit:  
:q
- If you haven't saved, you'll get an error,  
not a prompt to provide a file name
- To exit without saving:  
:q!

# More Command Mode

- Save, and then quit

`:wq`

- Put vi in the background, and start a new shell in the foreground

`:sh`

- You can find further goofiness online

# Advanced Normal Mode

- cut
  - In visual mode, use `d` instead of `y`
- Delete a line  
`dd`
- Copy the current line  
`yy`
- Undo the last Normal Mode command  
`u`

# Advanced Normal Mode

- Delete the current character

x

- Delete the current word

dw

- Transpose current and next char

xp

- Go into Ansert Mode at the end of the line

A



# Advanced Normal Mode

- Open new line above the current line in Insert Mode
  - `O (big oh)`
- Open new line below the current line in Insert Mode
  - `o (little oh)`
- Delete the rest of the current sentence
  - d)
- Delete the current char and enter Insert Mode
  - s

# Advanced Normal Mode

- Join two lines

```
firstlineALLONELINEfirstline
ALLONELINE
secondlineALLONELINEsecondli
neALLONELINE
~
```

- Hit J

```
firstlineALLONELINEfirstline
ALLONELINEsecondlineALLONEL
INEsecondlineALLONELINE
~
```

# Advanced Normal Mode

- You can do something multiple times, or for a distance greater than one by adding numbers:
- `5dd`
  - Delete the five lines starting at the current cursor position

# Advanced Normal Mode

- You can do something multiple times, or for a distance greater than one by adding numbers:
- `d55 <enter>`
  - Delete fifty five lines starting at the current cursor position

vim demo

End