

Why Text Editors?

• In traditional UNIXworld, everything is a text file (source code, configuration files, e-mail, input to text formatting programs, etc., etc.), so mastering a cryptic but powerful "text editor" can pay off.

Slide 2

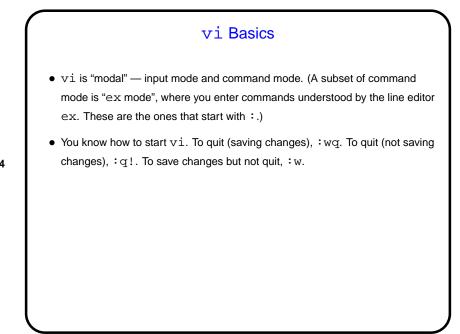
• Does this approach still make sense? Maybe, though you have to choose your other tools carefully to get maximum payoff. But a determined person can use the same text editor to write programs, compose e-mail messages, "word process", etc.)

Which Text Editor?

 Traditionally a "religious war" topic, with vi and emacs having the most supporters. Both very powerful and very widely available. There are others, but they're not as widely available, and often are more novice-friendly than expert-friendly.

Slide 3

- vi (or one of its clones) slightly more universally available. Plain vi is lightweight but a little primitive. vi under Linux is really vim, and has lots of extra features. Useful to know which are not "real" vi in case you ever have to use real vi. :set cp makes vim behave almost like "real" vi.
- emacs is almost as available and highly customizable can do almost anything (compile and test programs, send e-mail, etc.) from within it. (If I had it to do over again, I might well choose emacs!)

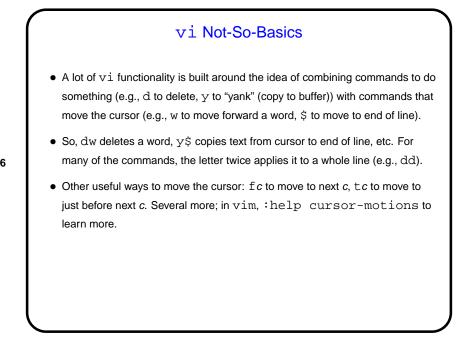


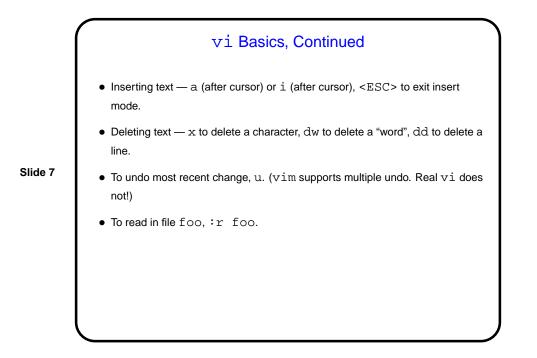


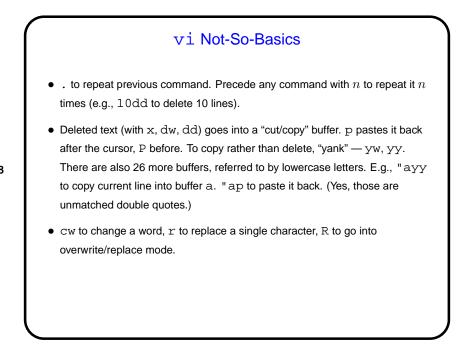
To move around, arrow keys usually work (and in vim you can use them in insert mode). Old way — which always works, but requires command mode — h, j, k, l. Does anyone still use those keys? Fanatical touch typists, maybe!

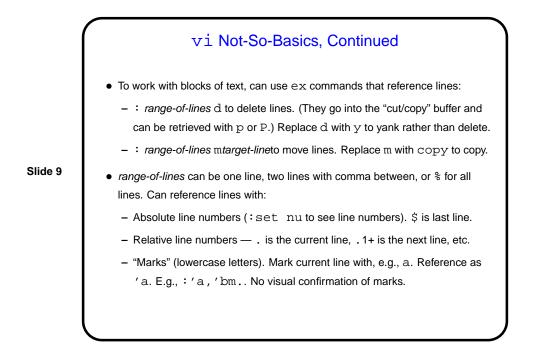
Slide 5

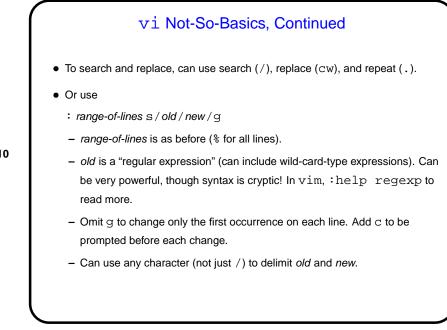
- Scrolling up and down ctrl-F and ctrl-B. Moving to start or end of line ^ and \$.
- Many other "cursor-movement" commands, e.g., w (next word) which can be usefully combined with commands to do something (next slide).
- To find foo, /foo<CR>. (<CR> means "enter" here.) Repeat with /<CR> (forward) or ?<CR> (backward), or n to repeat search in same direction.

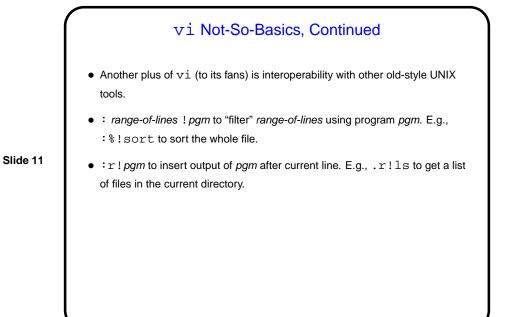


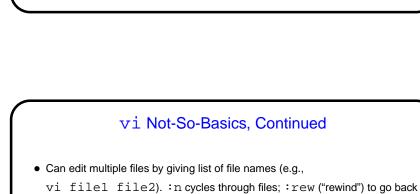






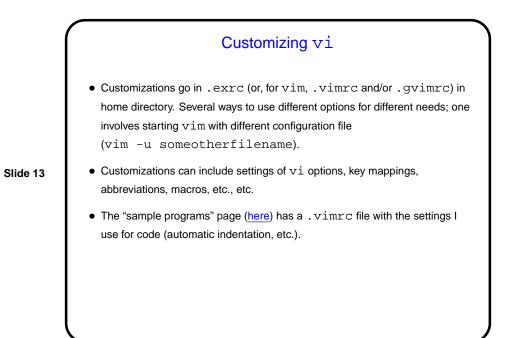


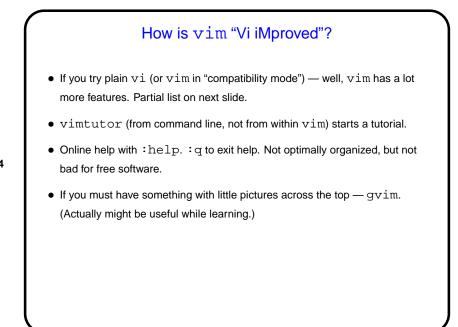


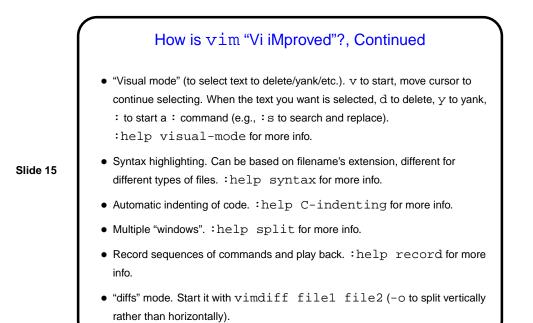


to first. This allows making similar changes in several files, or cutting and

pasting text from one file to another.







EMACS emacs is (IMO) the other major player in the text-editor wars. May be more powerful and customizable overall. Some other programs (e.g., bash) use some of the same key bindings. Add-ons available to do — "everything"? Maybe! (Try <ESC>-x doctor. ctrl-x ctrl-c to quit.) Add-ons/customization are done with Lisp code (similar to Scheme). Online help available — ctrl-H. ctrl-H T starts a tutorial. If you must have something with little pictures across the top — actually these days emacs started in a graphical environment has that. If you want the old-style text-only interface, use the command-line switch -nw. (There is also xemacs, but it's a different code base.)

More Unsolicited Advice

 Both vim and emacs are powerful editors and may be worth the trouble to learn — unless you plan to do all or most of your editing with programs that have their own editor. If nothing else, they will show you a different way of doing things! My advice is to try both and see if one of them appeals to you.

Slide 17

• As with other UNIX things, a good way to learn them is incrementally — learn a few things, practice them, then learn a few more. The online help/tutorials are good sources of new things to try. So is your local expert. A good approach is to think of something you do often and find tedious, and try to find a way to make it easier / faster.

Minute Essay • What text editor do you currently use under Linux? What do you like/dislike about it? Slide 18