Lecture 3

Editors: pico, emacs, and vi

COP 3344 Introduction to UNIX

Fall 2007

Acknowledgment: These slides are modified versions of Prof. Sudhir Aggarwal's slides

Editors

- Common text editors
  - *vi* (vee-eye)
    - Available on almost all Unix machines
    - Fairly powerful and sophisticated
  - *emacs* (ee-macs)
    - Also widely available
    - Powerful and popular
  - *pico*
    - Easier to learn but simpler and not as powerful
- Note: The caret (^) indicates that you must press and hold the control key first, then press the desired key
  - `^c` means press `control` key, then press `c`, then release `c`, then release `control`

CTRL C

Starting *pico*

- Run *pico*
  - `$ pico` works on an empty buffer
  - `$ pico file1` works on file
- Basic commands
  - Arrow keys are used to navigate around the document
  - Typing will insert text at the point of the cursor
  - Some available commands are at the bottom of the *pico* window
    - `^o` writes out the text to a file (a prompt will let you specify the name)
    - `^x` exits *pico*

Cutting and Pasting in *pico*

- ^^ (ctrl-*shift*-^) begins marking text at the current cursor position
- Use the arrow keys to mark text
- ^k cuts text (*k*ills)
- ^u then brings the text back at the current cursor position

pico Command Summary

`(arrows)`	Move cursor
`(bksp)`	Move cursor left one space, deleting character
`^a`	Move to beginning of line
`^b`	Move back one character (same as left arrow)
`^e`	Move to end of line
`^f`	Move forward one character (same as right arrow)
`^n`	Move to next line (same as down arrow)
`^p`	Move to previous line (same as up arrow)
`^v`	Move forward one page
`^y`	Move back one page
`^(space)`	Move to next word

More *pico* Commands

`^c`	Shows current position
`^d`	Delete character at current position
`^g`	Display help file (^V and ^Y to scroll through)
`^h`	Delete previous character (same as bksp)
`^i`	Insert TAB character (same as tab)
`^j`	Justify paragraph
`^^`	Begin selecting text at current cursor position
`^k`	Cut selected text
`^o`	Output current buffer to a file (save)
`^r`	Insert text from a file
`^u`	Undelete last line, series of lines, or marked block you deleted
`^w`	Search file for text
`^x`	Exit pico

emacs

- To use *emacs* remotely with windows:
  - Download and run the program X-Win32
    - Then `ssh -X` *remote-machine*
- Run emacs
  - `emacs -nw` without windows
  - `emacs` with windows
    - `^h t` to run tutorial
  - `emacs [-nw] <filename>` to open a file
  - `^x ^s` saves the buffered text
  - `^x ^c` exits `emacs`
- *emacs* can be customized using LISP

Basic *emacs* Commands

- Arrow keys can be used to navigate the document
  - Alternatively
    - `^b` moves *b*ack one character
    - `^f` moves *f*orward one character
    - `^n` moves to *n*ext line
    - `^p` moves to *p*revious line
    - `^v` moves to next page
    - `ESC v` moves to previous page ESC v
  - Mouse can be used without `-nw`
- `^g` quits a command

Cutting and Pasting in *emacs*

- `^k` cuts text (*k*ills) from cursor to end of line
- `^y` pastes text (*y*anks back)
- `^@` sets a *mark* at the current cursor position
  - Move cursor to the end of the text you want to cut
  - `^w` cuts text from *mark* to current cursor position
  - Move to the position where you want to paste
  - `^y` then brings the text back at the current position

emacs Command Summary

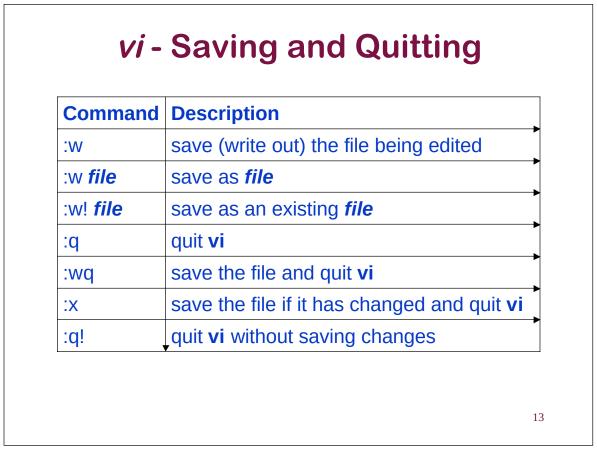
`(arrows)`	Move cursor
`(bksp)`	Move cursor left one space, deleting character (this may or may not work depending on your configuration)
`(del)`	Typically works like you would expect bksp to work
`^a`	Move to beginning of line
`^b`	Move back one character (same as left arrow)
`^e`	Move to end of line
`^f`	Move forward one character (same as right arrow)
`^n`	Move to next line (same as down arrow)
`^p`	Move to previous line (same as up arrow)
`^v`	Move forward one page

More *emacs* Commands

`^x ^s`	Save buffered text to currently specified file
`^x ^w`	Write buffered text to a specific file
`^x ^f`	Find a file and copy it into buffer
`^x ^c`	Exit *emacs*
`^d`	Delete character at current position
`^s`	Search forward
`^r`	Search backward
`^k`	Cut text from cursor to end of line
`^@`	Set mark
`^w`	Cut text from mark to current cursor position
`^y`	Paste text at current cursor position
`^g`	Cancel command

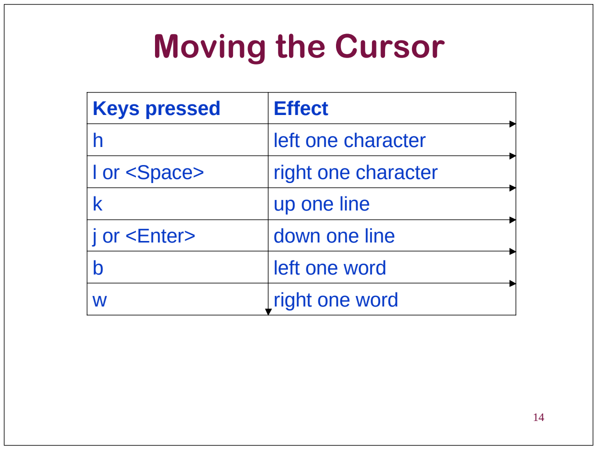
vi

- *vi* is available on all Unix systems
  - Run as `vi <filename>`
- *vi* has two modes
  - Command mode: in this mode characters you type are interpreted as commands
  - Insert mode: characters you type are inserted as part of the text
  - *vi* starts out in insert mode
  - Typing *i* switches to *i*nsert mode
  - The *ESC* key puts you back in command mode
- Commands are not displayed on the screen and do not require a return or enter



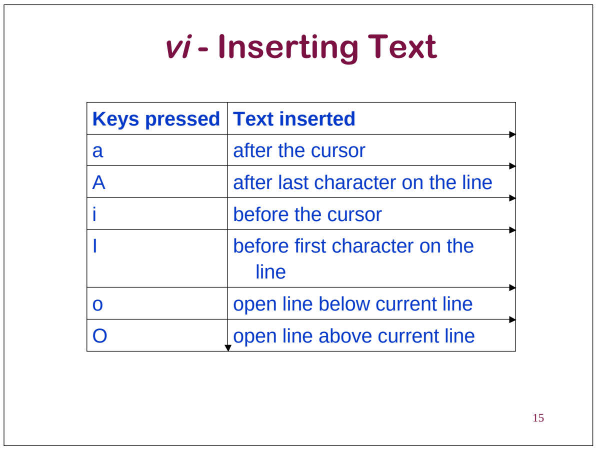
vi - Saving and Quitting

Command	Description
:w	save (write out) the file being edited
:w ***file***	save as ***file***
:w! ***file***	save as an existing ***file***
:q	quit **vi**
:wq	save the file and quit **vi**
:x	save the file if it has changed and quit **vi**
:q!	quit **vi** without saving changes



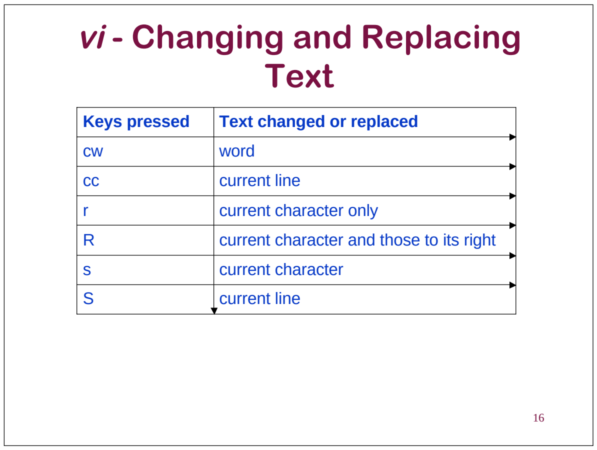
Moving the Cursor

Keys pressed	Effect
h	left one character
l or <Space>	right one character
k	up one line
j or <Enter>	down one line
b	left one word
w	right one word



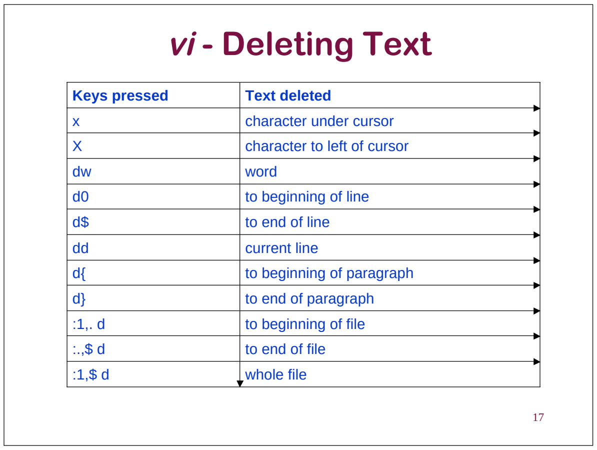
vi - Inserting Text

Keys pressed	Text inserted
a	after the cursor
A	after last character on the line
i	before the cursor
I	before first character on the line
o	open line below current line
O	open line above current line



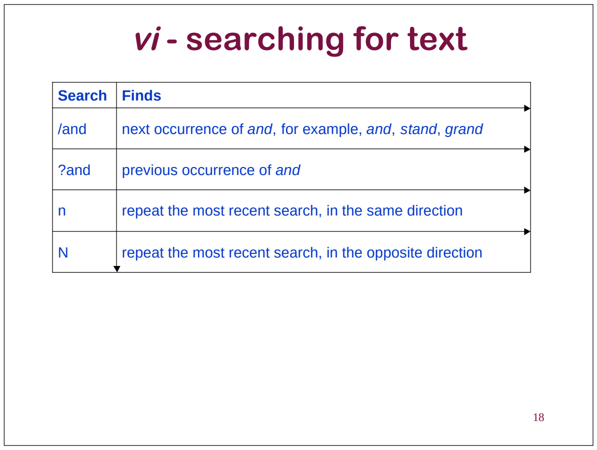
vi - Changing and Replacing Text

Keys pressed	Text changed or replaced
cw	word
cc	current line
r	current character only
R	current character and those to its right
s	current character
S	current line



vi - Deleting Text

Keys pressed	Text deleted
x	character under cursor
X	character to left of cursor
dw	word
d0	to beginning of line
d$	to end of line
dd	current line
d{	to beginning of paragraph
d}	to end of paragraph
:1,. d	to beginning of file
:.,$ d	to end of file
:1,$ d	whole file



vi - searching for text

Search	Finds
/and	next occurrence of *and*, for example, *and*, *stand*, *grand*
?and	previous occurrence of *and*
n	repeat the most recent search, in the same direction
N	repeat the most recent search, in the opposite direction