CISC 151
The UNIX Operating System

Lecture 4

Richard L. Holladay, CCNA, Ph.D.

Week 4

- Topics:
  - An Introduction to UNIX Text Editors
    - vi and vim
  - UNIX Electronic Mail
    - pico the editor for pine
    - pine

2

Text Editors

- Unix systems store both operating system configuration and machine configuration in plain text files
- A text editor is an essential tool for those who need to modify these files
- As more tools, such as linuxconf, are developed to maintain Linux system files, the need to manually edit these files is decreasing
- However, text editors are still used a lot!
- Need to bite the bullet and learn one (or two)

3

What's text?

- "Text" is defined as keyboard characters within the ASCII 7-bit character set, with a lowest value of '\x20' (decimal 32) -- the <space> character, and a highest character of '\x7e' (decimal 126) -- the ~ character.
- The only characters with lower values than <space> that are allowed in a true "text" file are:

```
<tab>  \t   '\x09'
<cr>   \r   '\x0d'
<lf>   \n   '\x0a'
<ff>   \f   '\x0c'
```

4

Unix Editors

- Editors in Unix come in two general flavors:
  - modal editors have "modes"
    - generally input mode and command mode
      - input mode allows entry of text
      - command mode allows positioning within the file and more sophisticated text modification
    - primary Unix examples: ed and vi

5

Unix Editors

- modeless editors (or WYSIWYG: What You See Is What You Get) have only one mode
  - positioning and text manipulation are done by special key sequences (like arrow keys and function keys)
    - could also be done by mouse actions or menus
  - primary Unix examples: emacs or pico

6

Console Text Editors

- **ed** - primitive line editor, sometimes invoked (like **ex**) in scripts
- **emacs** - screen based editor
- **pico** - screen based editor also used with pine
- **sed** - stream editor (filter program), like ed, for use in shell scripts/programs
- **vi** - visual (screen based but a GUI) interface for the **ex** editor
- **vim** - vi improved – Used with Linux

7

GUI (X-Windows) Text Editors

- **gedit**
  - (click on Gnome-footprint > programs > applications > gedit)
- **gless**
  - (click on Gnome-footprint > programs > utilities > Text File Viewer)
- **kedit**
  - (click on Gnome-footprint > KDE menus > applications > Text Editor)
- **kwrite**
  - (click on Gnome-footprint > KDE menus > applications > Advanced Editor)

8

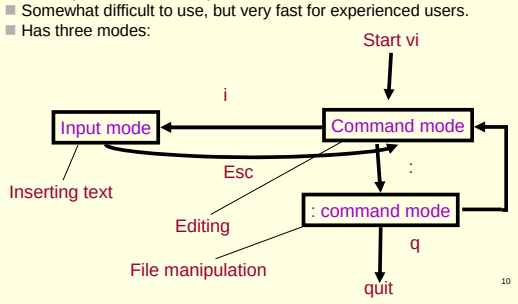
ed

- **ed**, like vi, is found on almost any Unix or Linux system
- Roots traced to teletypes first used before terminals
- Unlike vi, **ed** shows only one line of text at a time
- It's literally a "line" editor.
- For most purposes, ed has been replaced in normal usage by full-screen editors
- **vi** encapsulates **ed** inside its larger set of screen-oriented commands

9

vi

- Developed at UC Berkeley and comes with all versions of Unix.
- Somewhat difficult to use, but very fast for experienced users.
- Has three modes:



10

vi

- **vi** (**v**isual **i**nterface) editor is an interactive editor that is used to create and/or modify ASCII (American Standard Code for Information Interchange) text files.
- Character based
- Uses a screen display, but no mouse capability (Windows weenies beware!)
- All editing with the vi editor is done within a memory buffer.
  - Changes can be written to the disk or discarded.
- vi works only with text files and does not have any formatting capabilities.

11

weeny, n.[1] (From the OED)

1. *colloq.* A very young child.
1844 C. Ridley *Let.* Oct. (1958) xv. 180, I must tell you now about the chicks [i.e. children]. Little weeny is growing visibly. **1973** 'D. Halliday' *Dolly & Starry Bird* vi. 82 Not Mr. Paladrini who was so nice to the weenies? **1977** *Ottawa Citizen* 29 June 7/1 Our five-year-old granddaughter keeps asking when the trip is going to begin. Travelling with weenies is something that Mama and I have done for most of our lives.
2. *U.S. slang.* **a.** A girl; an effeminate man. **b.** An objectionable person.
1929 [see phooey *int. (n.)*]. **1963** *Amer. Speech* XXXVIII. 171 Some of the less frequent+phrases [for an effeminate young man] are: *dink*+and *weenie*. **1964** *Ibid.* XXXIX. 118 Free variation+is also established in the interchangeability of the names of smaller animals for socially unacceptable persons: *toad, squirrel*, and *shrimp* all serve for the zoologically unsound but all-inclusive *weenie*. also weanie, weany

12

The Oxford English Dictionary, 2nd Edition (1989)

vim

- Most releases of Linux actually use **vim** (vi improved), which has enhanced support for color modes with terminals, among other new features
- vi is now a symbolic link to vim

13

vi Modes

Command mode	Initial default mode for creating and editing files, cursor positioning and modification of existing text. All commands are initiated from this mode. Pressing Esc from either of the other modes will return you to command mode.
Entry mode	Used for entry of new text. Entering an insert command such i (insert), a (append) and o (open new line) will take you from command mode to entry mode. Pressing escape (Esc) returns to command mode. Entry commands are entered by themselves without pressing the Enter key.
Last-line mode	Used for saving your work and quitting vi. Type a colon (:) to get to this mode. Pressing the Enter key returns to command mode.

14

Command Mode

- This is the initial mode for creating and editing files. When you start the vi editor and open a file, you are in **command** mode. In this mode, you can enter **cursor positioning commands** to move around the file and **editing commands** to perform functions and modifications of existing text.
- You can change to **entry** mode or **last-line** mode from here. To move from entry mode to **last-line** mode or vice versa, you must first switch back to **command** mode.
- All commands are initiated from **command** mode.

15

Command Mode

- The Escape key always puts you in **command** mode. Use the Escape key if you are not sure what mode you are in. When in doubt, press Esc to return to command mode before issuing any other commands. While in command mode, you can move your cursor around the text on the screen and do advanced editing commands.

16

Cursor Positioning Commands

Command	Meaning
j or down arrow	Move down one line
k or up arrow	Move up one line
Space bar	Move right (forward) one character
w	Move forward one word (including punctuation)
b	Move back one word (including punctuation)
$	Move to end of line
0 (zero) or^	Move to beginning of line
Return	Move down to beginning of next line
h	Moves back one character
l	Move forward one character

17

Basic Text Deletion Commands

Command	Meaning
x (lowercase)	Delete character at the cursor
dw	Delete word (or part of word to right of cursor)
3dw	Delete three words
dd	Delete line containing the cursor
3dd	Delete three lines

18

Entry Mode

- You must be in entry mode to enter text. To enter text, you must type a vi insert command such as **i (input)**, **o (open new line)**, or **a (append)**. This takes vi out of command mode and puts it into entry mode. In this mode, keypresses will not be interpreted as editing commands. When you finish entering text in your file, press the Escape key to return to command mode.

19

Append, Insert & Open Commands

Command	Meaning
a	Append text after the cursor
A (Upper case)	Append text at the end of the line
i	Insert text before the cursor
o	Open a new line below the cursor

20

Last-line Mode

- From **command** mode, you can type a **:** which places you at the bottom line of the screen.
  - This is called **last-line** mode.
- The changes made to a document are kept in a memory buffer until written to the disk.
- From **last-line** mode, you can save your work periodically by writing it to disk while remaining in vi or you can save and quit vi.
- You can also quit vi without saving the changes.
- **Last-line** mode can also be used to search for strings and to set preferences.

21

Saving Files and Quitting vi

- Several **last-line** mode commands are available for saving your work and exiting vi. You should save your work to disk periodically using the **:w** (write) command. This writes the file to disk and leaves you in vi.
- To save a copy of the file currently being edited to disk under a new name, use the **:w** command followed by a new file name. This is a good way to make a backup of the file.
- To save the file and quit, you can use either the **:wq** or **ZZ** commands.

22

Saving Files and Quitting vi

- If you have ***accidentally*** made changes and want to ***discard*** them without saving or you want to exit vi and ensure that you have not made any changes, use the **:q!** (quit) command.
- If you open a read-only file and wish to modify it, you can save changes and overwrite the existing file with the **:wq!** (write-quit) command

23

Save and Quit Commands

Command	Meaning
:w	Write buffer (Save changes and continue working in vi)
:w new_filename	Write buffer to new_filename
:wq	Write buffer (save changes) and quit vi
ZZ (Upper case)	Save changes and quit vi. Alternative to :wq
:q!	Quit without saving changes
:wq!	Write buffer (save changes) and quit vi (The ! will override read only permissions if you are the owner of the file.)

4

Editing Session Commands

Command	Meaning
:set nu	Show line numbers
:set nonu	Hide line numbers
:set showmode	Display current mode of operation (turned on automatically if the vedit command is used)
:set noshowmode	Turn off mode display
:set	Display all vi variables set
:set all	Display all possible vi variables and their current settings

25

vi "visual" commands

Cursor movement (command mode)

- Ctrl-d Down (forward) half a page
- Ctrl-f Forward one page
- Ctrl-u Up half a page
- Ctrl-b Back one page

- G Go to end of file (capital G)
- 1G Go to beginning of file
- *n*G Go to line *n*

26

vi "visual" commands

The vi "cursor" keypad:

```
        k
    h       l
        j
```

In command mode

h left one character
l right one character
k up one line
j down one line

27

vi "visual" commands

Cursor positioning in command mode

^	Go to first character of current line
$	Go to last character of current line
w	Beginning of next word
e	End of next word
(	Backward one sentence
)	Forward one sentence

28

emacs

- Emacs: (Editor MACroS)
  - developed by Richard Stallman and James Gosling among many others
  - modeless
  - has versions for Unix, Windows, and other systems
  - menu-driven and mouse-driven under X-windows.

29

emacs

- Emacs uses special keys (ESC and CTRL) to perform editor functions other than input
- This editor can do <u>everything</u>
  - Contains a complete programming language (a LISP interpretter) which can be used to write functions for use in the editor
- Not enough time this week to cover it

30

Learning vi(m)

- Many good vi websites
  https://engineering.purdue.edu/ECN/Resources/KnowledgeBase/Docs/20020202121609
- Do the vim tutorial by typing **vimtutor** (at a shell prompt near you)
- Download vim for windows from:
  www.vim.org Also a lot of great info regarding vim here!
- Do something useful while wasting your time in Windows!

31

pine Email - Topics

- The pico (**pi**ne **co**mposer) editor
- What is pine email?
- Accessing pine email
- Basic email functions
  - Checking Messages
  - Composing Messages
  - Replying to Messages
  - Forwarding Messages
- Creating address book entries
- Adding an email signature
- Working with Attachments

pico

- pico (pine composer)
- pico is a very simple text editor found on some Unix and Linux systems (like Suse)
- It is the easiest, but least powerful text editor in common use on Linux and Unix systems
- It is a component of the Pine email client which uses it to compose email messages
- All the commands use the control key
- Pine and Pico are not supported in Red Hat releases later than 7.x, for licensing reasons
- To start : pico *filename* or use pico *–w filename* to prevent word-wrapping

33

pico

- Commands displayed at bottom of screen, and context-sensitive help is provided
- **pico** is modeless
- As characters are typed they are immediately inserted into the text
- Editing commands are entered using control-key combinations
- As a work-around for communications programs that swallow certain control characters, you can emulate a control key by pressing ESCAPE twice, followed by the desired control character, e.g. "ESC ESC c" would be equivalent to entering a ctrl-c.

34

pico

- Five basic features:
  - paragraph justification
  - searching
  - block cut/paste
  - spelling checker
  - file browser
- Paragraph justification (or filling) takes place in the paragraph that contains the cursor, or, if the cursor is between lines, in the paragraph immediately below.

35

pico

- Paragraphs are delimited by blank lines, or by lines beginning with a space or tab.
- String searches are not sensitive to case. A search begins at the current cursor position and wraps around the end of the text.

36

pico

- The most recent search string is offered as the default in subsequent searches.
- Some ^ Commands:
  - ctrl-g Get Help screen
  - ctrl-x Exit (save if buffer changed)
  - ctrl-o Write file (out), don't exit
  - ctrl-r Read file into current cursor position
  - ctrl-v Next page on screen
  - ctrl-u Previous page (up) on screen
  - ctrl-p Previous line
  - ctrl-n Next line

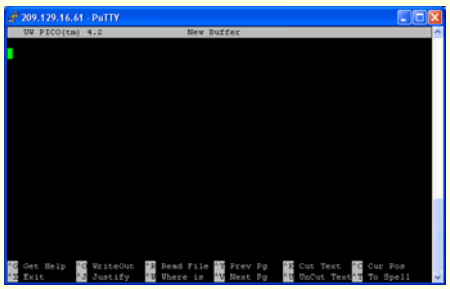
37

pico

- Blocks of text can be moved, copied or deleted with creative use of the command for mark (ctrl-^), delete (ctrl-k) and undelete (ctrl-u)
- The delete command will remove text between the "mark" and the current cursor position, and place it in the "cut" buffer. The undelete command effects a "paste" at the current cursor position.
- The spell checker examines all words in the text.
- Spell checking can be cancelled at any time.

38

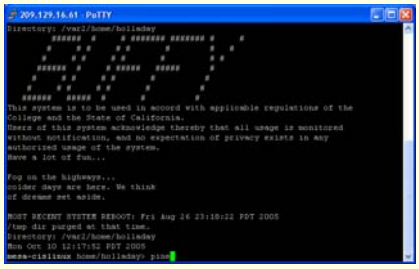
pico



39

Accessing pine Email

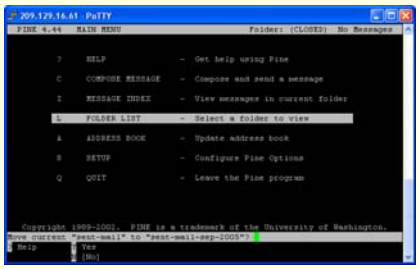
- Type *pine* and press Enter



40

Pine Main Menu

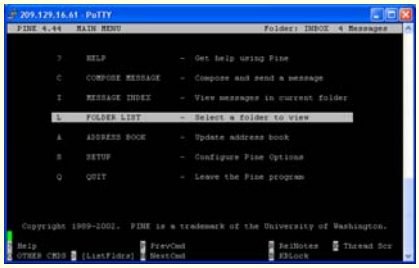
- Answer the question at the bottom by selecting “y” or “n” and press Enter



41

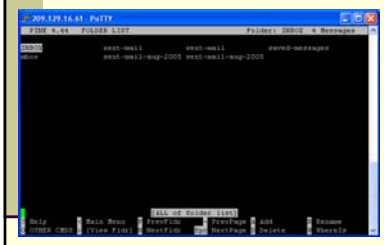
Checking Messages

- To check your email go to your “Inbox”, or “Message Index”, by pressing I, or your Folder List by pressing L.



42

Checking Messages

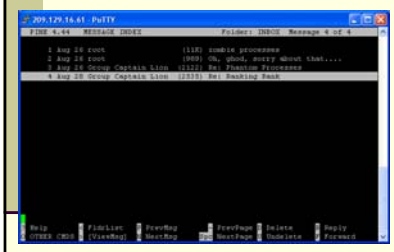


- If you press L, this screen appears with a list of your folders.
- Email accounts start with three folders.
  - "INBOX" holds your current messages
  - "sent-mail" stores copies of every message you send
  - "saved-messages" is were you can save messages you want to keep

- Open a folder by pressing Enter while the folder name is highlighted.
- The highlight may be moved with the arrow keys.

43

Checking Messages

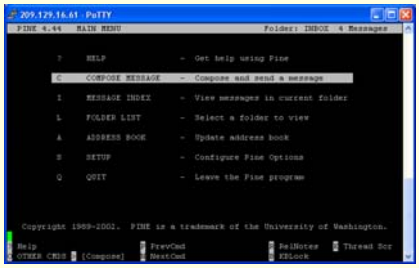


- If you press I, this screen appears with a list of your messages. This screen also appears when you open the Inbox folder.
- Open messages the same way you opened folders.
- Some commands for this screen:
  - D deletes the selected message; U undeletes the selected message
  - S saves the selected message
  - R opens a reply
  - F starts a forwarded message

- Pressing M will bring you back to the Main Menu

Composing New Messages

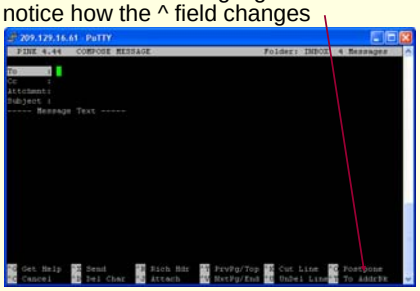
- Select *Compose Message* from the main menu



45

Composing New Messages

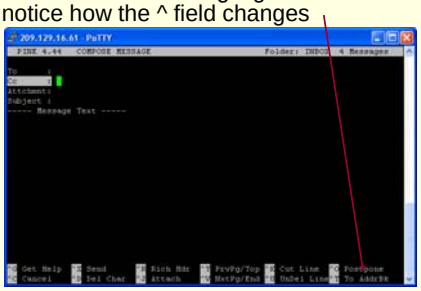
- Move the cursor to highlight each field and notice how the ^ field changes



46

Composing New Messages

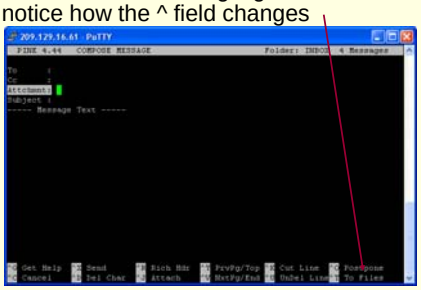
- Move the cursor to highlight each field and notice how the ^ field changes



47

Composing New Messages

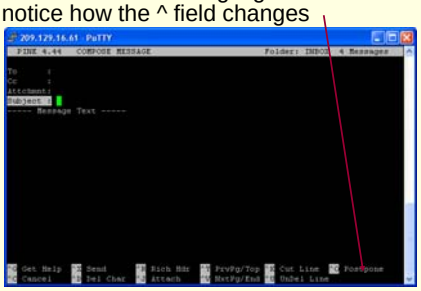
- Move the cursor to highlight each field and notice how the ^ field changes



48

Composing New Messages

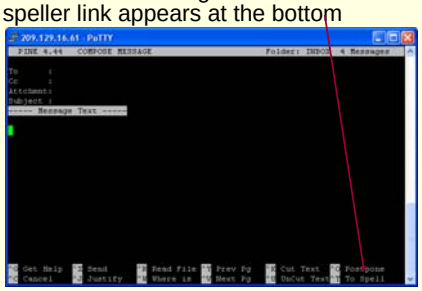
- Move the cursor to highlight each field and notice how the ^ field changes



49

Composing New Messages

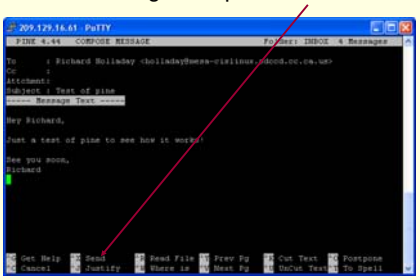
- Move to the Message Text area and a check speller link appears at the bottom



50

Composing New Messages

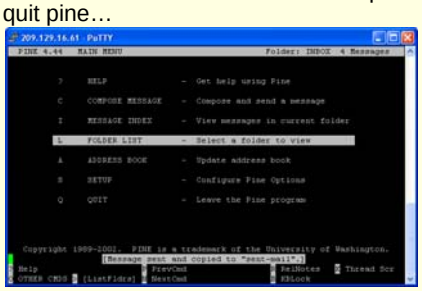
- Write a message and press ^x to send



51

Composing New Messages

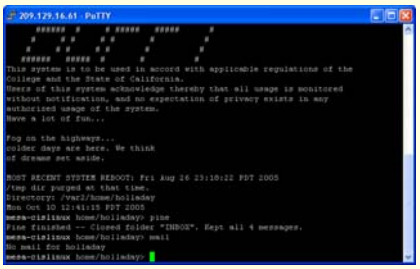
- You're returned to the Main Menu…press q to quit pine…



52

Checking for mail…

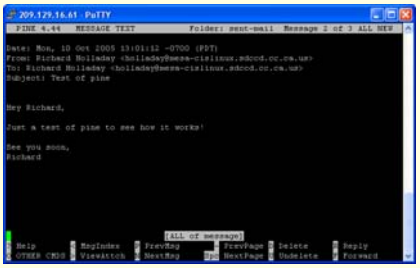
- Just enter *mail* at the shell prompt



53

Replying

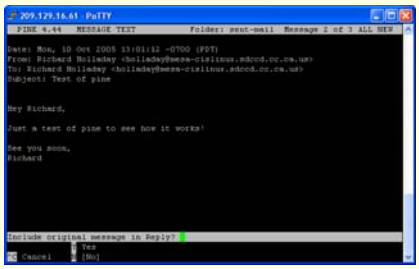
- While viewing, press R (no cntl) to reply…



54

Replying

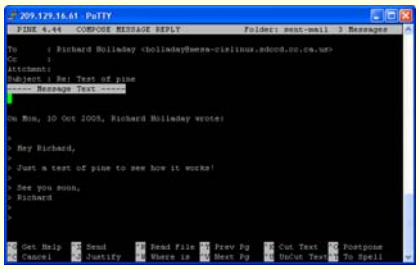
- Answer the prompt..



55

Replying

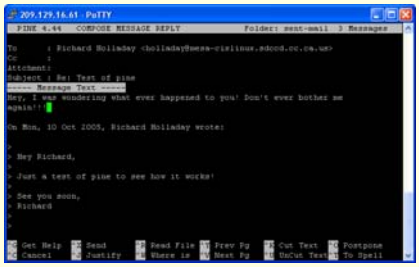
- …and enter your reply at the cursor…



56

Replying

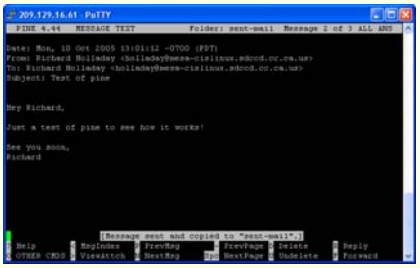
- …and press ^S to send your reply.



57

Replying

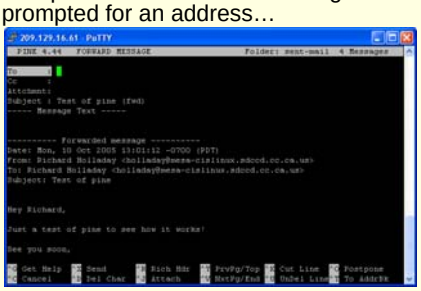
- you're taken back to the message you were viewing.



58

Forwarding

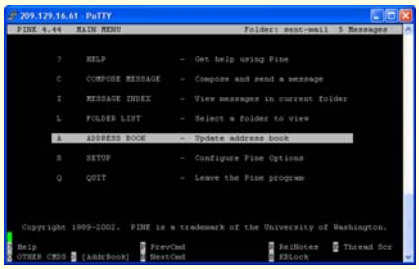
- Just press F to forward a message. You're prompted for an address…



59

Creating an Address Book Entry

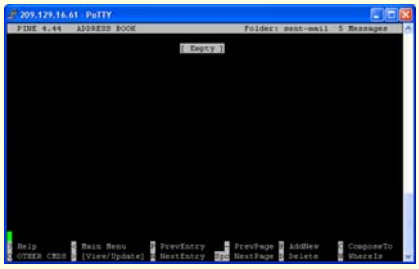
- From the Menu press A…



60

Creating an Address Book Entry

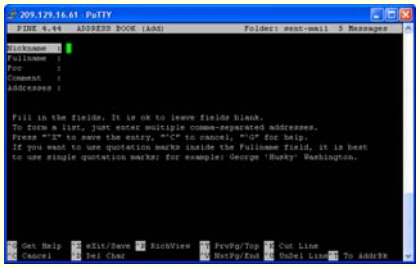
- Press shift 2 (the @ sign) to add a new entry



61

Creating an Address Book Entry

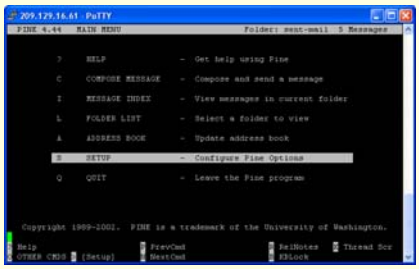
- Press ^X to save the entry…



62

Adding an Email Signature

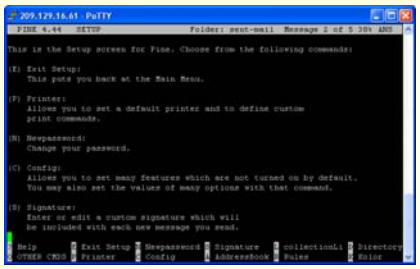
- Go to the Setup menu



63

Adding an Email Signature

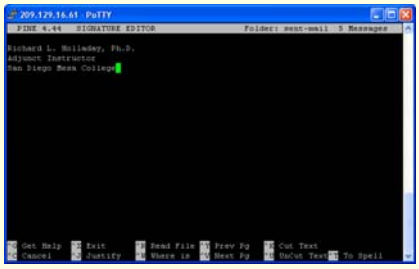
- Press S again to enter Signature…



64

Adding an Email Signature

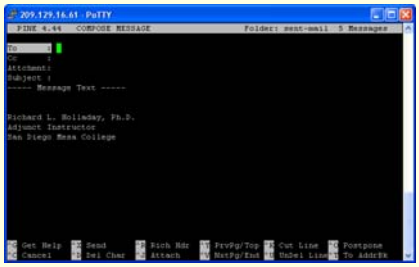
- Type your signature and press ^X to save and exit



65

Adding an Email Signature

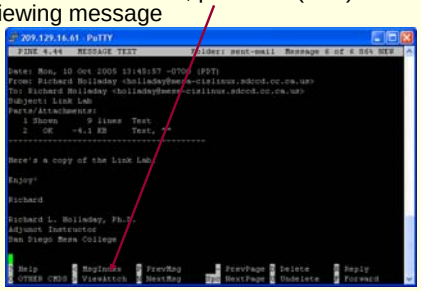
- Check it out with a new Compose Message…



66

Viewing an Attachment

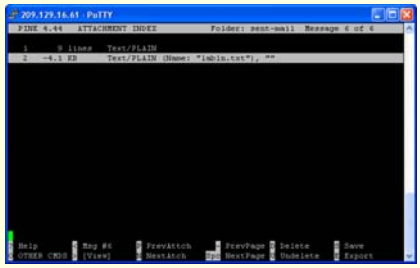
- To view attachment, press >V (or V) while viewing message



67

Saving Attachments

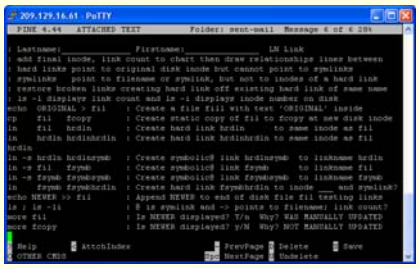
- press V again (or Enter) …



68

Saving an Attachment

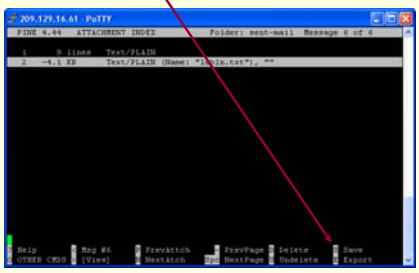
- Use spacebar to view or < to return back



69

Saving an Attachment

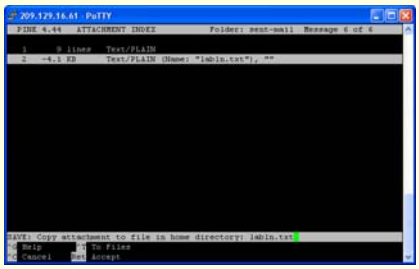
- To save, press S while viewing Attachment Index



70

Saving an Attachment

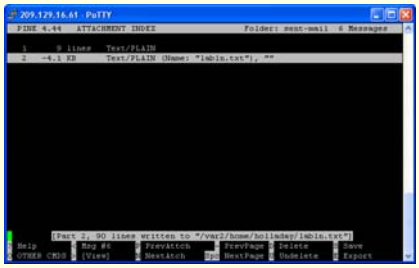
- Press enter…



71

Saving an Attachment

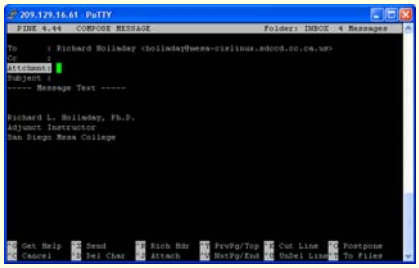
- The prompt will give you the path where it is saved



72

Sending an Attachment

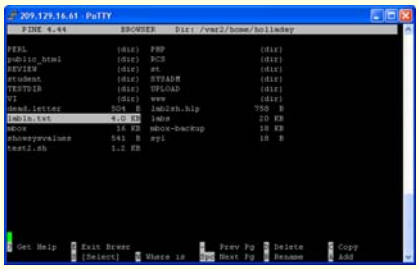
- While composing an email, press ^T for your file list



73

Sending an Attachment

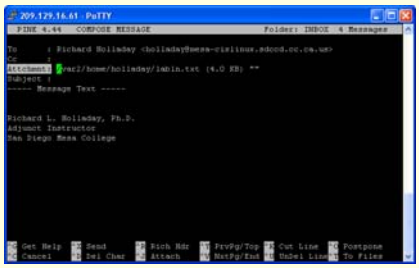
- Highlight file and press S to Select



74

Sending an Attachment

- Continue writing and press ^X to send



75

Next Week

This Week

- Do the vim tutorial either in UNIX or download vim for Windows and do it there

Next Week

- Midterm
  - Chapters 1, 2; 6 – 11
  - Lecture screen content & vim tutorial
- Lecture:
  Chapter 12: Networking with TCP/IP

76