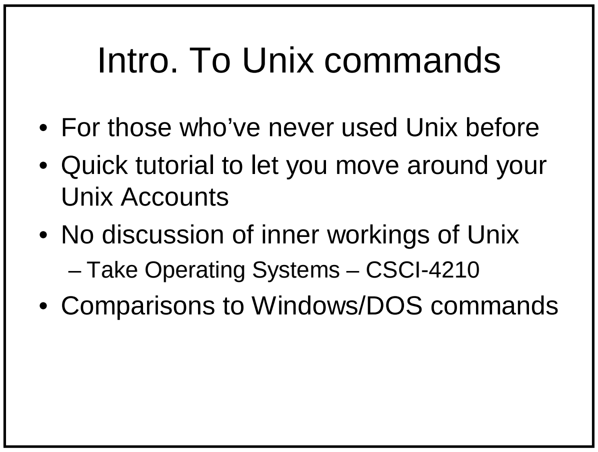
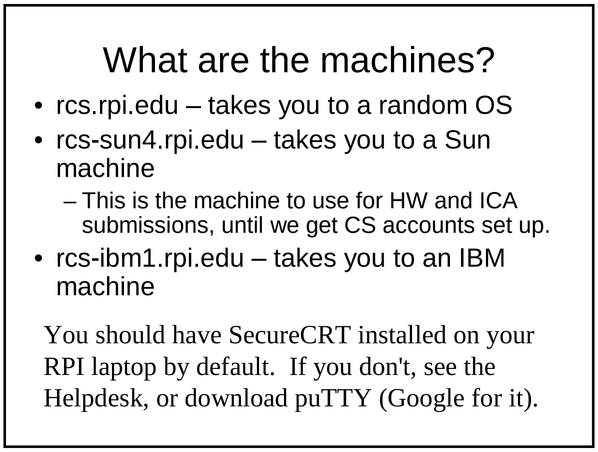


Intro. To Unix commands

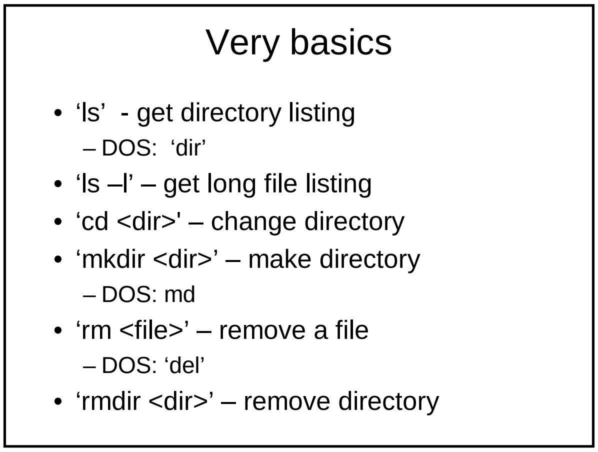
- For those who’ve never used Unix before
- Quick tutorial to let you move around your Unix Accounts
- No discussion of inner workings of Unix
  - Take Operating Systems – CSCI-4210
- Comparisons to Windows/DOS commands



What are the machines?

- rcs.rpi.edu – takes you to a random OS
- rcs-sun4.rpi.edu – takes you to a Sun machine
  - This is the machine to use for HW and ICA submissions, until we get CS accounts set up.
- rcs-ibm1.rpi.edu – takes you to an IBM machine

You should have SecureCRT installed on your RPI laptop by default. If you don't, see the Helpdesk, or download puTTY (Google for it).



Very basics

- ‘ls’ - get directory listing
  - DOS: ‘dir’
- ‘ls –l’ – get long file listing
- ‘cd <dir>' – change directory
- ‘mkdir <dir>’ – make directory
  - DOS: md
- ‘rm <file>’ – remove a file
  - DOS: ‘del’
- ‘rmdir <dir>’ – remove directory

More basics

- ‘more <file>’ – contents of a file
  - DOS: ‘type’
- ‘cp <from_path> <to_path>’ – copy a file
  - DOS: ‘copy’
- ‘mv <from_path> <to_path>’ – move a file
  - DOS: ‘move’

chmod

- Change mode (permissions) of file/directory
- using ‘ls –l’, 10 fields of information are shown
- ex: drwxrw-r—
- first position: ‘d’ (directory) or ‘-’ (file)
- next three: user permissions:
  - ‘r’ – Read permissions
  - ‘w’ – Write permissions
  - ‘x’ – eXecute permissions
- next three: group permissions
- last three: world permissions

chmod continued

- permissions are represented as octal numbers
- **`rwxrwxrwx`** = 111 111 111 = 777
- **`rwxrw-r--`** = 111 110 100 = 764
- **`rw-------`** = 110 000 000 = 600
- etc.
- **`chmod <mode> <file>`**
- ex: **`chmod 755 script.plx`**
  - Give yourself all permissions, group and world read and execute permissions

User Friendly chmod

- Can specify modes without using octal representations.
- Still a 3 character code.
  - First char: u (user), g (group), o (other)
  - Second: + (add permissions) or - (remove permissions)
  - Third: r (read), w (write), x (execute)
- chmod u+x file.pl
  - give yourself execute permissions on file.pl
- chmod o+r file.pl
  - give others (ie, world) read permissions on file.pl

Windows vs Unix

- They don't like each other.
- Many problems can (and will) arise due to difference in end-of-line character.
  - Unix: \n
  - Windows: \r\n
  - In Unix, a Windows \r can show up as ^M
  - In Windows, a Unix \n (missing the \r) can show up as a □ and no newline (at least in Notepad – Wordpad is mildly smarter)
- To change a file from Windows to Unix, use the command:
  - dos2unix
  - Ex: **`dos2unix oldfile > newfile`**
  - unix2dos also exists
- If you *ever* save a file in Windows and then transfer it to Unix, you should run dos2unix

Transferring files

- To get a file from your Windows PC to your Unix RCS account:
- Use an ftp client (ex: CuteFTP) or the default Windows ftp program
  - Ask Paul for help with this if you don't know ftp
  - ftp address of RCS is ftp.rpi.edu
  - Make sure you transfer in ASCII mode, not Binary!!
- Or use an SCP client (ex: WinSCP)
  - Download from http://winscp.sourceforge.net/
  - Connect to rcs.rpi.edu
- Avoid simply connecting to your network drive via Samba – this automatically transfers in binary mode, and you will need to run dos2unix

Creating a file in Unix

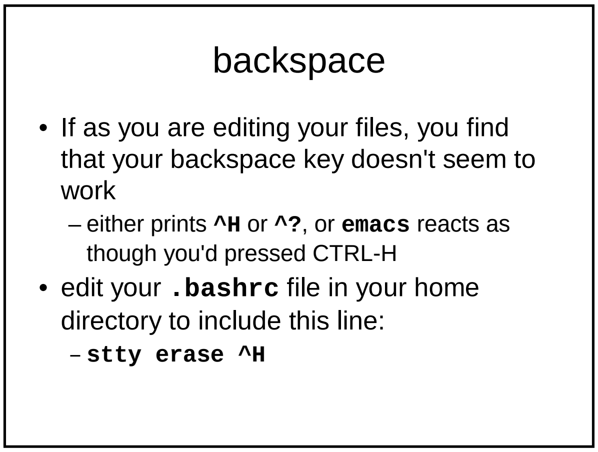
- Many options. Two most common: **`emacs`** and **`vi`** (“vee-eye”, not “vye” or “six”)
  - Which is ‘better’ is a cause of online holy wars
- Many **many** tutorials online to help you with either program
- There is an emacs reference card on the Definitions & Links page
- A third possibility is **`pico`**. This may be significantly easier for you if you've never edited a file on unix before.
  - All commands are shown on bottom of pico window
- To start a new file or edit an existing file, type name of editor followed by filename:
  - **`emacs file.pl`**
  - **`vi file.pl`**
  - **`pico file.pl`**

Beginning emacs shortcuts

- CTRL-X CTRL-F – open new or existing file
- CTRL-X S – prompt to save file
- CTRL-X CTRL-S – save file without prompt
- CTRL-X W – save file as
- CTRL-A – beginning of line
- CTRL-E – end of line
- CTRL-X CTRL-J – jump to line #....
- CTRL-K – Cut to end of line
- CTRL-Y – Paste most recent cut
- ESC-X – many mini-buffer commands (replace, undo, search, etc (tab-complete for list))
- ESC-> – end of file
- ESC-< – beginning of file
- CTRL-G – quit mini-buffer (if you typo when typing a command)
- CTRL-X CTRL-C – Exit emacs

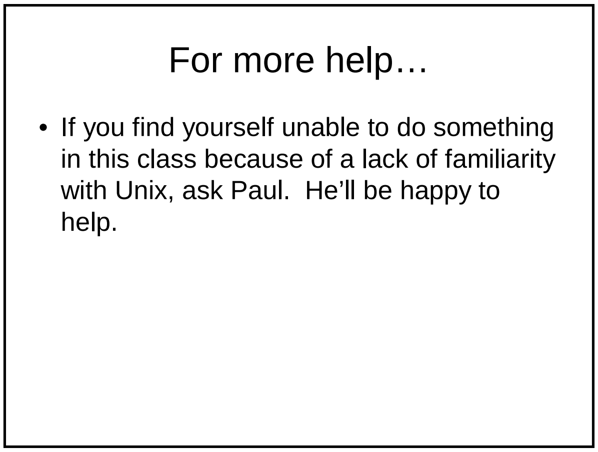
vi Commands

- ESC - leave insert/append/replace mode
- i - enter Insert mode at current position
- I - enter Insert mode line start
- a - enter Append mode at current position
- A - enter Append mode at line end
- r - replace single character at current position
- R - enter Replace mode at current position
- /foo<enter> - search forwards for "foo"
- ?foo<enter> - search backwards for "foo"
- w - move cursor to next word
- $ - move cursor to end of line
- ^ - move cursor to start of line
- x - delete current character
- dw - delete to end of word
- d$ - delete to end of line
- :w<enter> - write (save) file
- :wq<enter> - save file and quit
- :q<enter> - quit
- :q!<enter> - quit without saving changes



backspace

- If as you are editing your files, you find that your backspace key doesn't seem to work
  - either prints **`^H`** or **`^?`**, or **`emacs`** reacts as though you'd pressed CTRL-H
- edit your **`.bashrc`** file in your home directory to include this line:
  - **`stty erase ^H`**



For more help…

- If you find yourself unable to do something in this class because of a lack of familiarity with Unix, ask Paul. He’ll be happy to help.