A Brief Introduction to the AOS Classroom Computers and Linux/Unix – Oct 2022

Pete Pokrandt
AOS Systems Admin
Room 1447 AOSS
poker@aos.wisc.edu

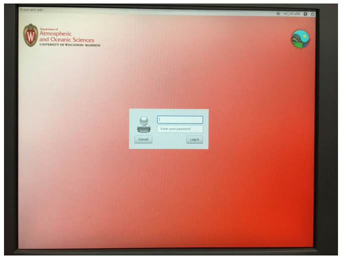
1

AOS classroom computers

- Dual boot – Windows 10 or CentOS 7 Linux – Linux is the default OS
- Please log off and be sure machines are back in Linux when you are done
- 15 machines in room 1411
  4 machines in room 1443
- JupyterHub server for remote python
  https://jupyterhub.aos.wisc.edu:1225

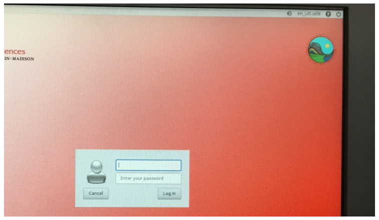
2

The Linux Login Screen


Atmospheric and Oceanic Sciences
UNIVERSITY OF WISCONSIN–MADISON
Enter your password
Cancel
Log In

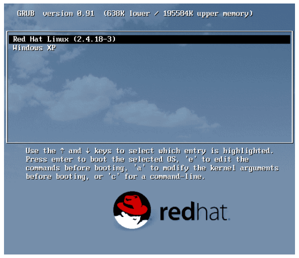
3

To switch from Linux to Windows



4

To switch from Linux to Windows



```
GRUB  version 0.91  (638K lower / 195584K upper memory)

Red Hat Linux (2.4.18-3)
Windows XP

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.
```

Use the DOWN arrow to select Windows, then press <Enter>

5

Windows Logon info

- Guest user is ‘aos’, password on the board
- aos account is local to each machine

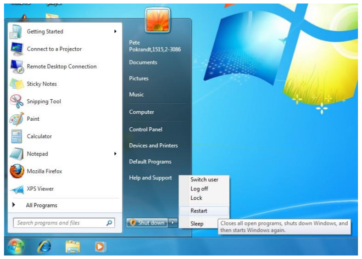
6

Software available under Windows in the 1411 lab

- □ Microsoft Office
- □ OpenOffice.org/LibraOffice
- □ Adobe Acrobat, Photoshop, Illustrator, Indesign
- □ Internet Explorer / Firefox / Chrome
- □ EdGCM, Hydra, miscellaneous others

7

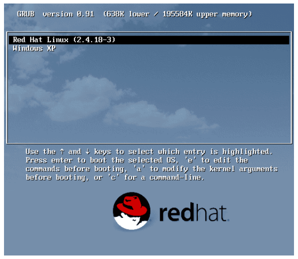
Reboot back to Linux



Closes all open programs, shuts down Windows, and then starts Windows again.

8

To switch from Windows to Linux

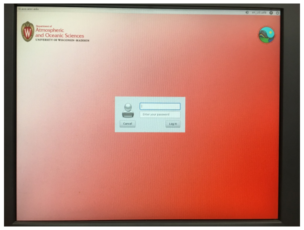


```
GRUB  version 0.91  (638K lower / 195584K upper memory)

Red Hat Linux (2.4.18-3)
Windows XP

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the
    commands before booting, 'a' to modify the kernel arguments
    before booting, or 'c' for a command-line.
```

Use the DOWN arrow to select Windows, then press <Enter>

9

The Linux Login Screen



'aos' user works on linux also.

Only local to each machine

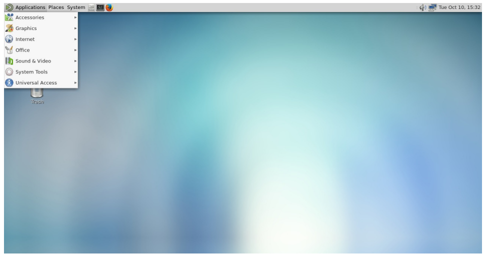
Use your username and password to log in – data on the server, common across all machines

10

Linux desktop

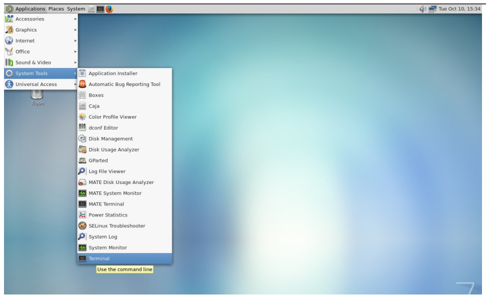
11

Linux desktop



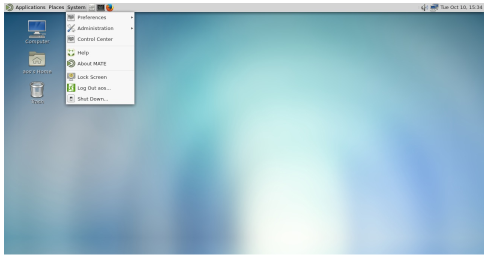
12

Linux – open a terminal window



13

Linux – Logging off of the machine



14

Brief Intro to Linux/Unix

- Operating Systems
- Brief History of Unix
- Basics of a Unix session
- The Unix File System
- Working with Files and Directories
- Your “Environment”
- Common Commands

15

Brief Intro to Unix (cont’d)

- Compilers, Email, Text processing
- Image Processing

- The ‘vi’ editor

16

Operating Systems

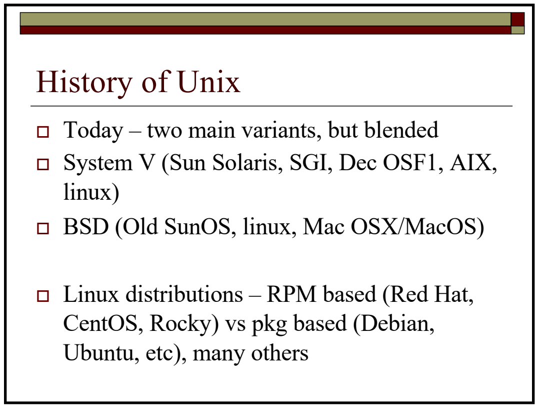
- The program that controls all other parts of a computer
- Familiar OS’s: MS Windows
  Mac OSX
  Unix/Linux variations
  Novell, VMS, OS/2,
  iOS (phones
  Android and tablets)

17

History of Unix

- Created in 1969 by Kenneth Thompson and Dennis Ritchie at AT&T
- Revised in-house until first public release 1977
- 1977 – UC-Berkeley – Berkeley Software Distribution (BSD)
- 1983 – Sun Workstations produced a Unix Workstation
- AT&T unix -> System V

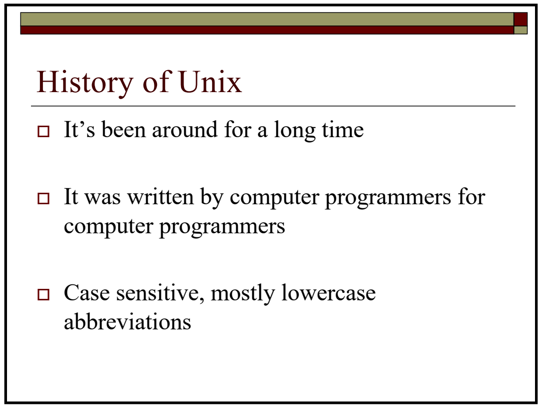
18



History of Unix

- Today – two main variants, but blended
- System V (Sun Solaris, SGI, Dec OSF1, AIX, linux)
- BSD (Old SunOS, linux, Mac OSX/MacOS)
- Linux distributions – RPM based (Red Hat, CentOS, Rocky) vs pkg based (Debian, Ubuntu, etc), many others

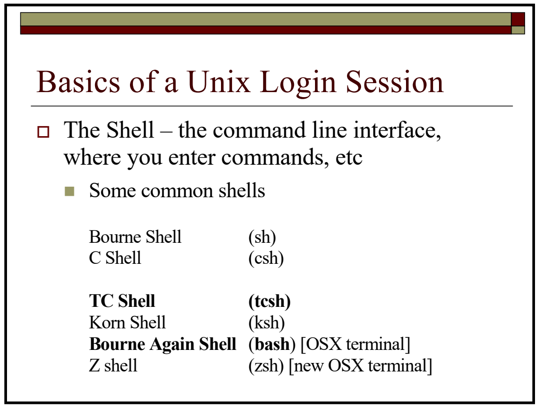
19



History of Unix

- It's been around for a long time
- It was written by computer programmers for computer programmers
- Case sensitive, mostly lowercase abbreviations

20

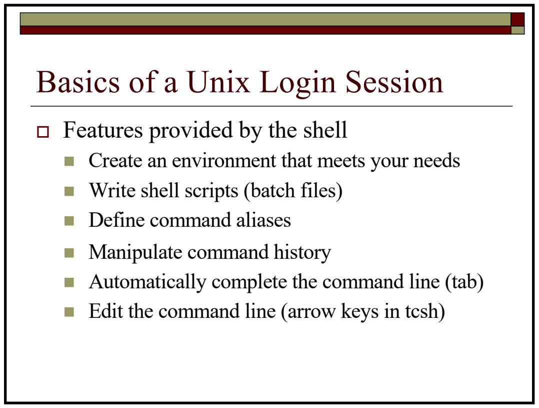


Basics of a Unix Login Session

- The Shell – the command line interface, where you enter commands, etc
  - Some common shells

Bourne Shell	(sh)
C Shell	(csh)
TC Shell	**(tcsh)**
Korn Shell	(ksh)
Bourne Again Shell	(**bash**) [OSX terminal]
Z shell	(zsh) [new OSX terminal]

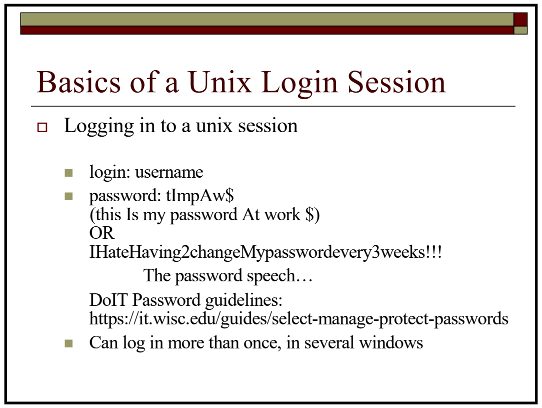
21



Basics of a Unix Login Session

- Features provided by the shell
  - Create an environment that meets your needs
  - Write shell scripts (batch files)
  - Define command aliases
  - Manipulate command history
  - Automatically complete the command line (tab)
  - Edit the command line (arrow keys in tcsh)

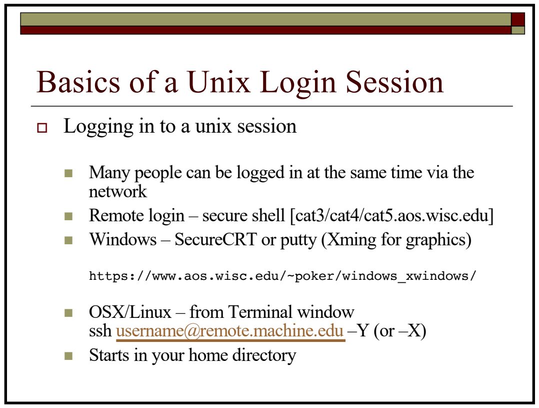
22



Basics of a Unix Login Session

- Logging in to a unix session
  - login: username
  - password: tImpAw$
    (this Is my password At work $)
    OR
    IHateHaving2changeMypasswordevery3weeks!!!
    The password speech…
    DoIT Password guidelines:
    https://it.wisc.edu/guides/select-manage-protect-passwords
  - Can log in more than once, in several windows

23

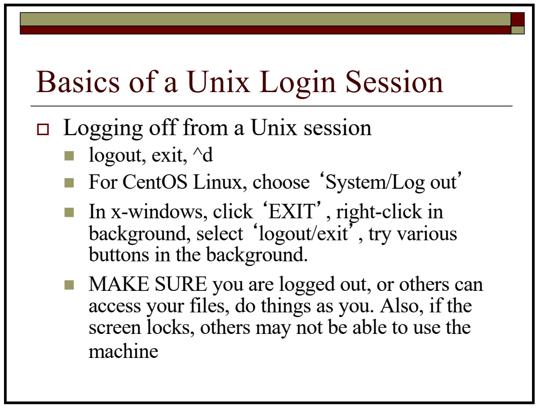


Basics of a Unix Login Session

- Logging in to a unix session
  - Many people can be logged in at the same time via the network
  - Remote login – secure shell [cat3/cat4/cat5.aos.wisc.edu]
  - Windows – SecureCRT or putty (Xming for graphics)

    `https://www.aos.wisc.edu/~poker/windows_xwindows/`
  - OSX/Linux – from Terminal window
    ssh username@remote.machine.edu –Y (or –X)
  - Starts in your home directory

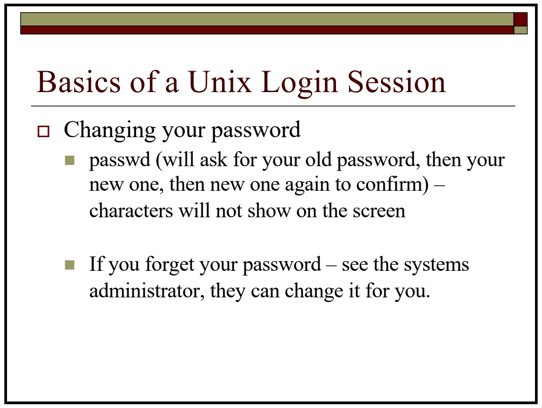
24



Basics of a Unix Login Session

- Logging off from a Unix session
  - logout, exit, ^d
  - For CentOS Linux, choose ‘System/Log out’
  - In x-windows, click ‘EXIT’, right-click in background, select ‘logout/exit’, try various buttons in the background.
  - MAKE SURE you are logged out, or others can access your files, do things as you. Also, if the screen locks, others may not be able to use the machine

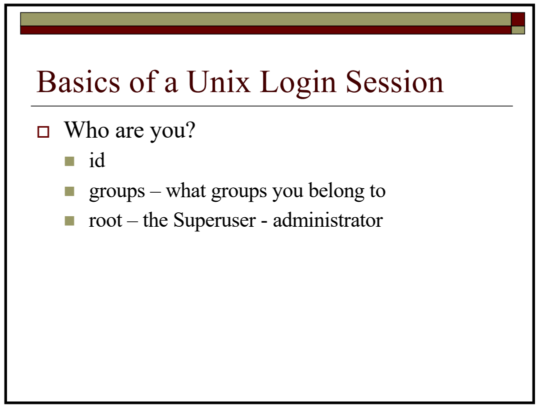
25



Basics of a Unix Login Session

- Changing your password
  - passwd (will ask for your old password, then your new one, then new one again to confirm) – characters will not show on the screen
  - If you forget your password – see the systems administrator, they can change it for you.

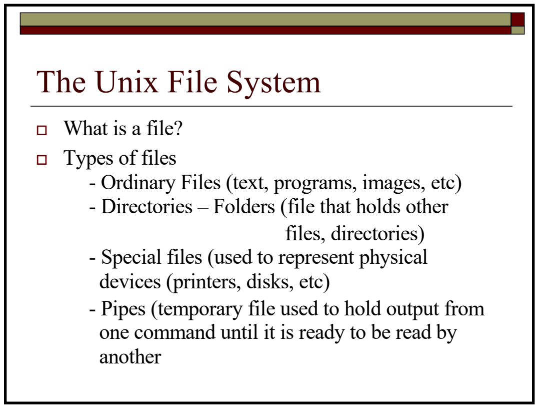
26



Basics of a Unix Login Session

- Who are you?
  - id
  - groups – what groups you belong to
  - root – the Superuser - administrator

27

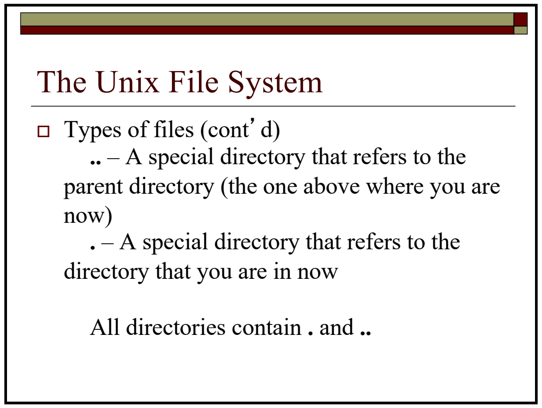


The Unix File System

- What is a file?
- Types of files

  - Ordinary Files (text, programs, images, etc)
  - Directories – Folders (file that holds other files, directories)
  - Special files (used to represent physical devices (printers, disks, etc)
  - Pipes (temporary file used to hold output from one command until it is ready to be read by another

28



The Unix File System

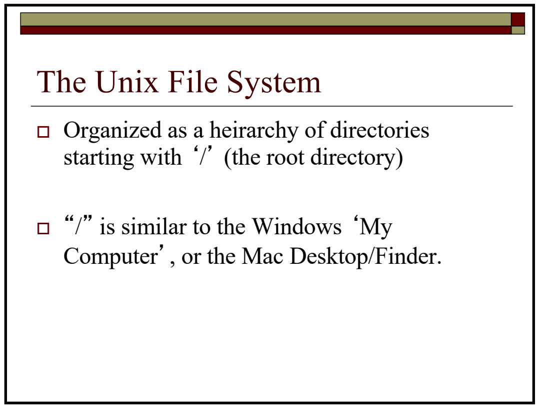
- Types of files (cont’d)

  .. – A special directory that refers to the parent directory (the one above where you are now)

  . – A special directory that refers to the directory that you are in now

  All directories contain **.** and **..**

29

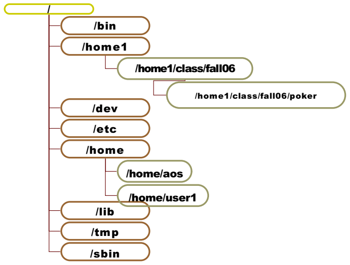


The Unix File System

- Organized as a heirarchy of directories starting with ‘/’ (the root directory)
- “/” is similar to the Windows ‘My Computer’, or the Mac Desktop/Finder.

30

The Unix File System



31

The Unix File System

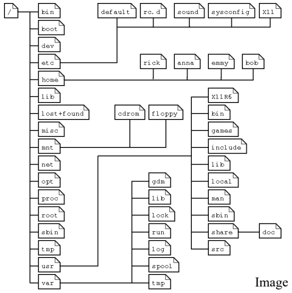


Image credit: tldp.org

32

The Unix File System

- □ Home Directory – where you are when you first log in (usually under /home - here, under /home1/class/fall18)
- □ Open Terminal starts in home or Desktop directory
- □ Current Working Directory (pwd)
- □ Absolute vs Relative Path Names

  /home1/class/fall18/poker/dir1/file1
  dir1/file1

33

The Unix File System

- □ Change directory to dir1 (cd command)

  cd dir1

  Ways to refer to that same file

  /home1/class/fall18/poker/dir1/file1
  file1
  ../dir1/file1
  ./file1

34

The Unix File System

- □ Common System Directories
  - □ / - root directory
  - □ /bin – common programs, shared by system/users
  - □ /boot – startup files, kernel, boot manager
  - □ /dev – references to peripheral hardware (disks, GPU)
  - □ /etc – administrative/configuration files/programs
  - □ /home – user home directories
  - □ /initrd – information for booting
  - □ /lib – libraries used by programs and languages

35

The Unix File System

- □ Common System Directories
  - □ /lost+found – files saved during power failures, etc
  - □ /misc – for miscellaneous purpouses
  - □ /mnt – standard mount point for external file systems
  - □ /net – standard mount point for remote file systems
  - □ /opt – third party software
  - □ /proc – virtual file system with info about system resources
  - □ /root – administrative user home directory (different than ‘root’ /

36

The Unix File System

- Common System Directories (cont’d)
  - /tmp – scratch area for temporary files
  - /usr – system files/directories shared by users
  - /var – variable/temp files (mail, printing, OS updates)
  - /usr/include – C include files
  - /home/aos– home directory for user ‘aos’
  - /tornado/home1/class/fall06/poker – home directory for user ‘poker’
  - /usr/local – locally added programs, libraries, etc
  - /usr/local/bin, /usr/local/lib, /usr/local/include, etc.

37

Using Unix Commands

- Case Sensitive! (ls not the same as Ls or LS)
- The Prompt: where you enter your commands

  `agnes[poker] %1` (csh, tcsh)
  `agnes$` (sh, ksh, bash)
  `agnes$` (root/admin shell)
- General command syntax

  `command [-flags] arg1 arg2`…

38

Using Unix Commands

- Use backspace or delete to correct errors

  `stty erase` [hit the key you want to use]
- Online manual pages for almost all commands

```
man man
man passwd
man -k compiler
```

39

Using Unix Commands

- Processes – unique process ID number for every process that is running
- Commands to identify processes
  - ps
  - ps –flu poker
  - ps –efl
  - ps –aux (bsd type systems)

40

Using Unix Commands

- ps –flu poker

```
# ps -flu poker
 F S      UID   PID  PPID  C PRI NI     ADDR     SZ    WCHAN
STIME TTY      TIME CMD

8 S    poker  1047     1  0  99 20 70cb0ec0    155 70cb0f2c
   Jan 16 ?        0:00 /bin/sh

8 S    poker  1049  1048  0  40 20 70ccd5f0    269 7015587a
   Jan 16 ?        0:00 /var/tmp/lm_TMW12.ld

8 S    poker  1048  1047  0  41 20 70ccceb8    133 70641c1c
   Jan 16 ?        0:00 sh -c while read line;
```

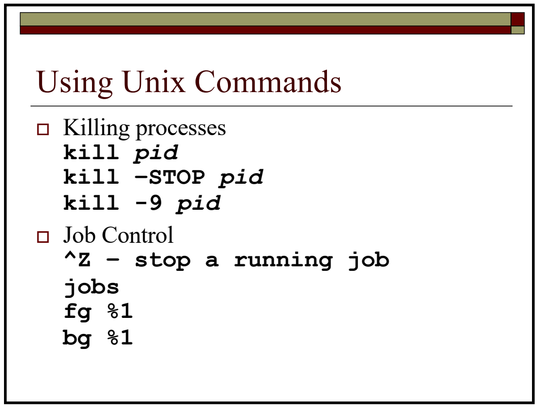
41

Using Unix Commands

- Jobs – per shell shortcut of programs running

```
agnes 27% jobs
[1] + Running          firefox
```

42



Using Unix Commands

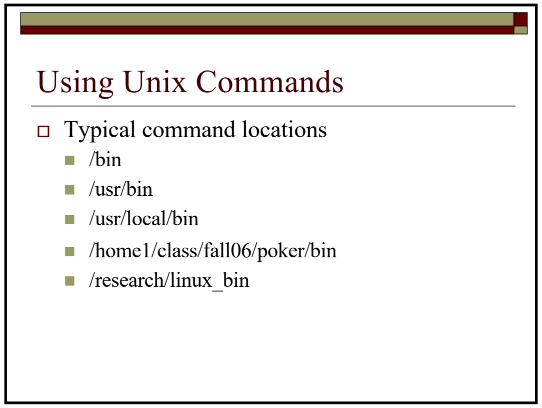
- Killing processes

```
kill pid
kill -STOP pid
kill -9 pid
```

- Job Control

```
^Z - stop a running job
jobs
fg %1
bg %1
```

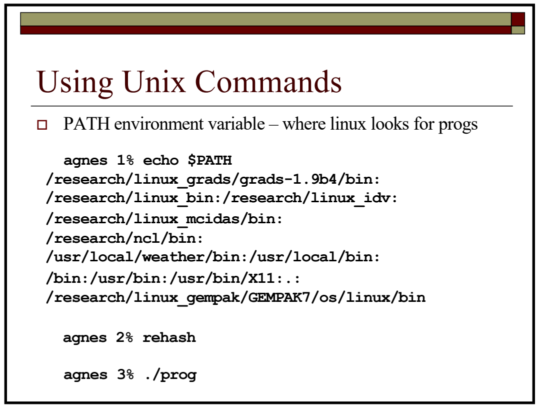
43



Using Unix Commands

- Typical command locations
  - /bin
  - /usr/bin
  - /usr/local/bin
  - /home1/class/fall06/poker/bin
  - /research/linux_bin

44

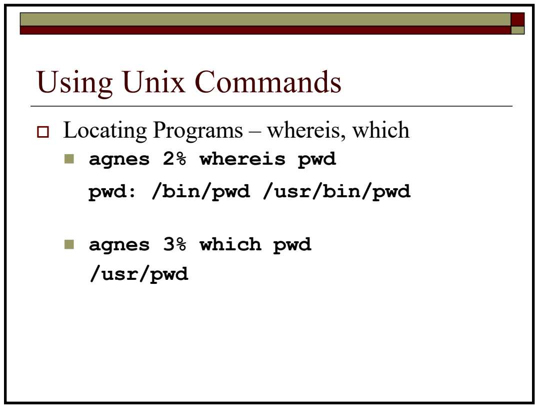


Using Unix Commands

- PATH environment variable – where linux looks for progs

```
  agnes 1% echo $PATH
/research/linux_grads/grads-1.9b4/bin:
/research/linux_bin:/research/linux_idv:
/research/linux_mcidas/bin:
/research/ncl/bin:
/usr/local/weather/bin:/usr/local/bin:
/bin:/usr/bin:/usr/bin/X11:.:
/research/linux_gempak/GEMPAK7/os/linux/bin

  agnes 2% rehash

  agnes 3% ./prog
```

45



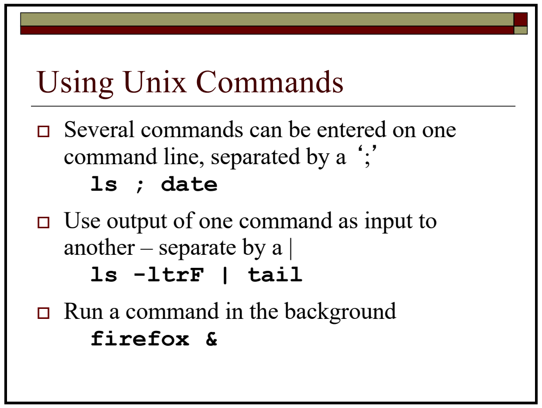
Using Unix Commands

- Locating Programs – whereis, which
  - `agnes 2% whereis pwd`

    `pwd: /bin/pwd /usr/bin/pwd`
  - `agnes 3% which pwd`

    `/usr/pwd`

46



Using Unix Commands

- Several commands can be entered on one command line, separated by a ';'

```
ls ; date
```

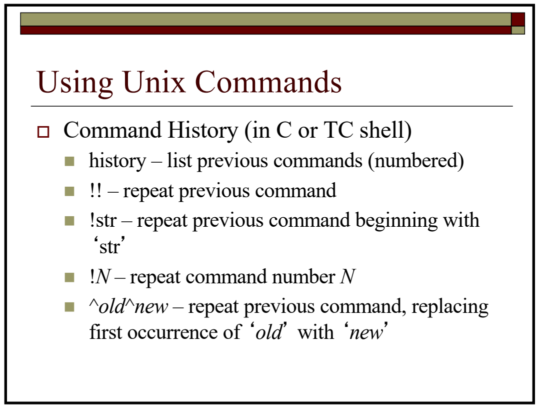
- Use output of one command as input to another – separate by a |

```
ls -ltrF | tail
```

- Run a command in the background

```
firefox &
```

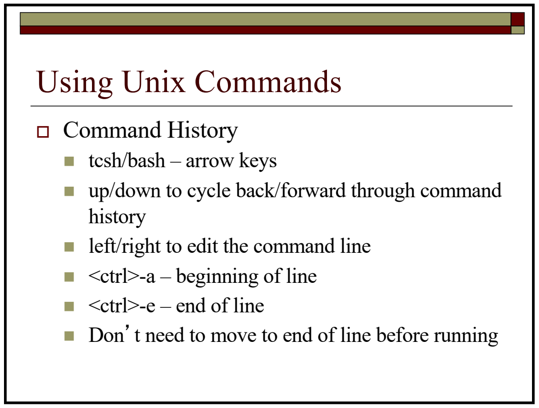
47



Using Unix Commands

- Command History (in C or TC shell)
  - history – list previous commands (numbered)
  - !! – repeat previous command
  - !str – repeat previous command beginning with 'str'
  - !*N* – repeat command number *N*
  - ^*old*^*new* – repeat previous command, replacing first occurrence of '*old*' with '*new*'

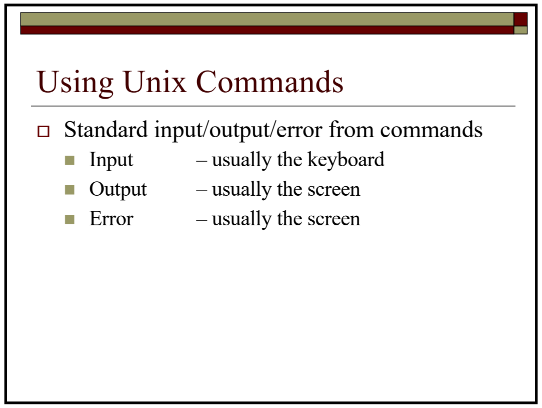
48



Using Unix Commands

- Command History
  - tcsh/bash – arrow keys
  - up/down to cycle back/forward through command history
  - left/right to edit the command line
  - <ctrl>-a – beginning of line
  - <ctrl>-e – end of line
  - Don’t need to move to end of line before running

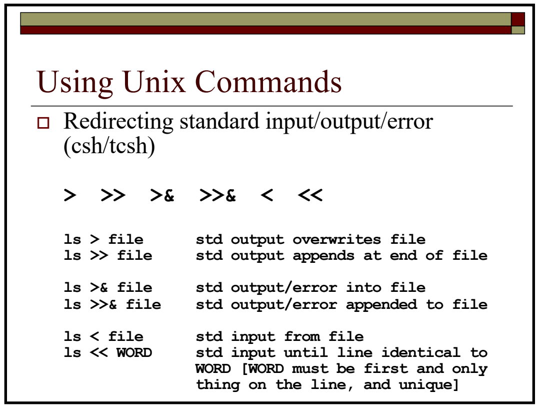
49



Using Unix Commands

- Standard input/output/error from commands
  - Input – usually the keyboard
  - Output – usually the screen
  - Error – usually the screen

50

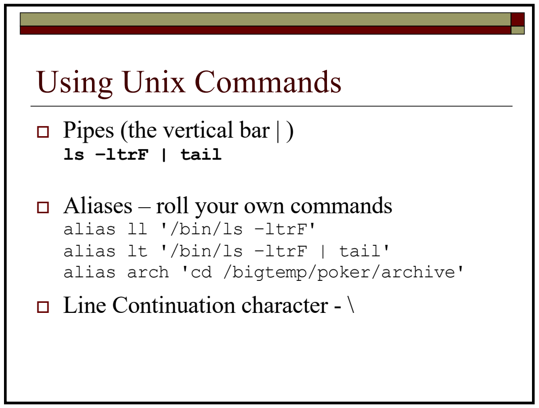


Using Unix Commands

- Redirecting standard input/output/error (csh/tcsh)

```
>  >>  >&  >>&  <  <<

ls > file       std output overwrites file
ls >> file      std output appends at end of file

ls >& file      std output/error into file
ls >>& file     std output/error appended to file

ls < file       std input from file
ls << WORD      std input until line identical to
                WORD [WORD must be first and only
                thing on the line, and unique]
```

51



Using Unix Commands

- Pipes (the vertical bar |)

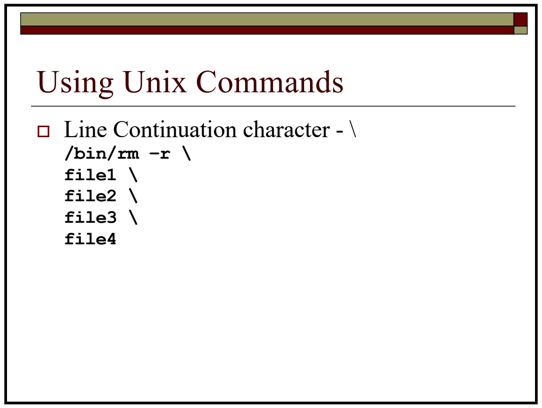
```
ls -ltrF | tail
```

- Aliases – roll your own commands

```
alias ll '/bin/ls -ltrF'
alias lt '/bin/ls -ltrF | tail'
alias arch 'cd /bigtemp/poker/archive'
```

- Line Continuation character - \

52

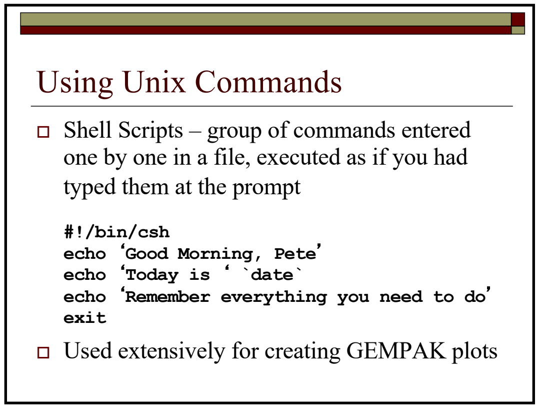


Using Unix Commands

- Line Continuation character - \

```
/bin/rm -r \
file1 \
file2 \
file3 \
file4
```

53



Using Unix Commands

- Shell Scripts – group of commands entered one by one in a file, executed as if you had typed them at the prompt

```
#!/bin/csh
echo ‘Good Morning, Pete’
echo ‘Today is ‘ `date`
echo ‘Remember everything you need to do’
exit
```

- Used extensively for creating GEMPAK plots

54

Working with Files and Directories

- Creating files
  - cat – concatenate files

```
cat > file1
this text will be put into file1
^D

cat file1 file2 file3 > file4

cat file1 file2 > file1
```

55

Working with Files and Directories

- echo – echo commands to stdout (the screen?)

```
echo ‘this text will be put into file1’ >
file1

echo ‘this text will be appended after
the last’ >> file1
```

- touch – create an empty new file, or update modification time of an existing file

```
touch file
```

56

Working with Files and Directories

- Editing files
  - What is a text editor compared to a word processor?
  - vi, nedit, gedit, nano, pico, emacs

```
vi file1
nedit file1 &
```

57

Working with Files and Directories

- Text editors
  - vi(m) - cryptic text editor included with all unix
  - nedit - graphical editor simliar to notepad
  - gedit - another graphical editor
  - pico/nano - nicer character based text editor
  - emacs- powerful, customizable text editor

58

Working with Files and Directories

- Displaying files
  - Cat – file scrolls up the screen
    `cat file1`
  - Pagers (more, less) – pause between screenfuls
    `less file1`
  - Text editors (vi, nedit, pico, emacs)
  - Head – displays the first 10 lines of a file
    `head -20 file` (first 20 lines)
  - Tail – displays the last 10 lines of a file
    `tail -f20 file` (the last 20 lines, then anything appended to file)

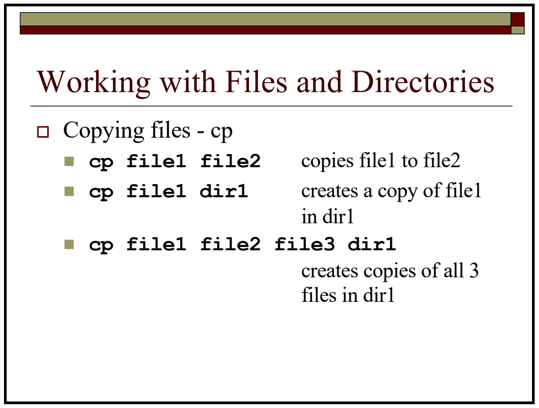
59

Working with Files and Directories

- Listing files – ls

```
ls
   a     b     dir1  file1

ls -a
   .     ..    a     b     dir1  file1

ls -l file1
-rw-r--r--  1 poker user 203 Jan 13 16:39 file1
```

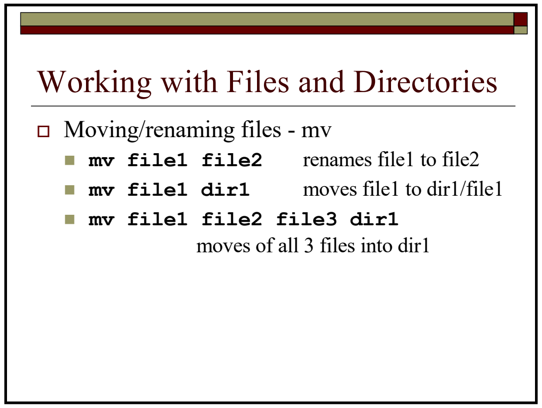
60



Working with Files and Directories

- Copying files - cp
  - **`cp file1 file2`** copies file1 to file2
  - **`cp file1 dir1`** creates a copy of file1 in dir1
  - **`cp file1 file2 file3 dir1`**
    creates copies of all 3 files in dir1

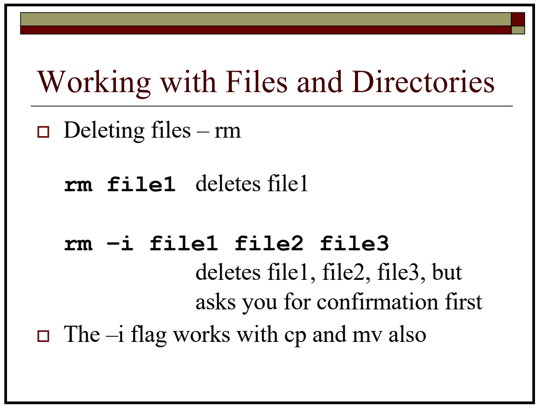
61



Working with Files and Directories

- Moving/renaming files - mv
  - **`mv file1 file2`** renames file1 to file2
  - **`mv file1 dir1`** moves file1 to dir1/file1
  - **`mv file1 file2 file3 dir1`**
    moves of all 3 files into dir1

62



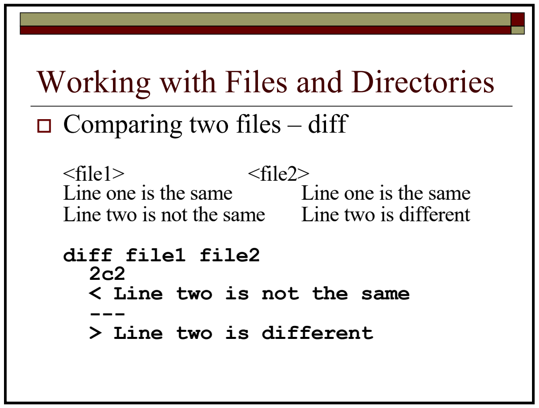
Working with Files and Directories

- Deleting files – rm

  `rm file1` deletes file1

  `rm –i file1 file2 file3`
  deletes file1, file2, file3, but
  asks you for confirmation first
- The –i flag works with cp and mv also

63



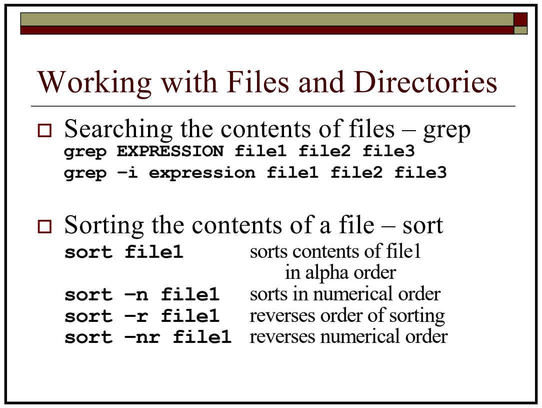
Working with Files and Directories

- Comparing two files – diff

<file1>	<file2>
Line one is the same	Line one is the same
Line two is not the same	Line two is different

```
diff file1 file2
  2c2
  < Line two is not the same
  ---
  > Line two is different
```

64



Working with Files and Directories

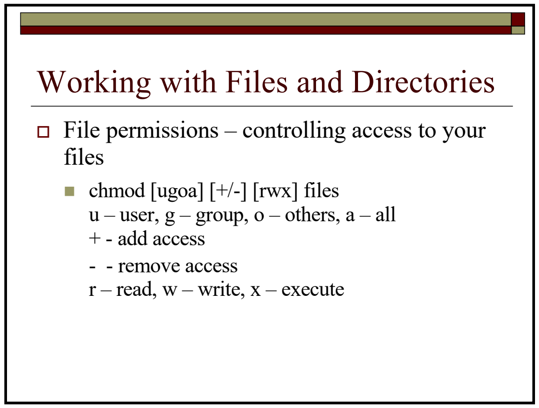
- Searching the contents of files – grep

```
grep EXPRESSION file1 file2 file3
grep –i expression file1 file2 file3
```

- Sorting the contents of a file – sort

`sort file1`	sorts contents of file1 in alpha order
`sort –n file1`	sorts in numerical order
`sort –r file1`	reverses order of sorting
`sort –nr file1`	reverses numerical order

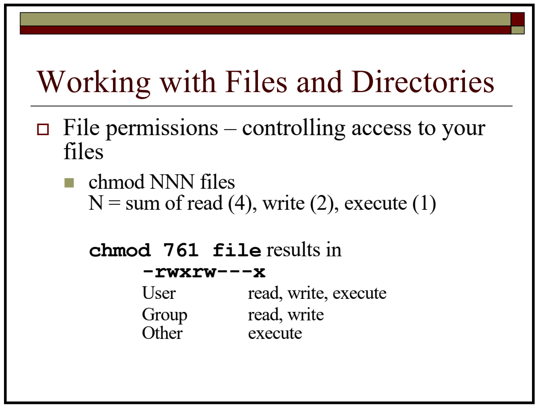
65



Working with Files and Directories

- File permissions – controlling access to your files
  - chmod [ugoa] [+/-] [rwx] files
    u – user, g – group, o – others, a – all
    + - add access
    - - remove access
    r – read, w – write, x – execute

66



Working with Files and Directories

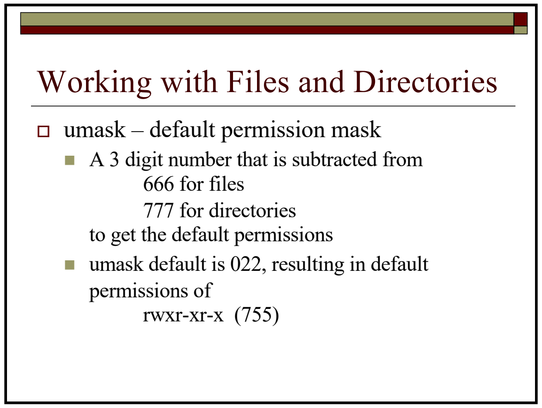
- □ File permissions – controlling access to your files
  - chmod NNN files
    N = sum of read (4), write (2), execute (1)

    `chmod 761 file` results in

    `-rwxrw---x`

    User read, write, execute
    Group read, write
    Other execute

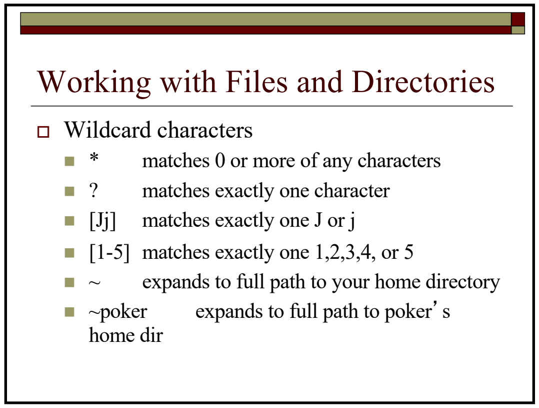
67



Working with Files and Directories

- □ umask – default permission mask
  - A 3 digit number that is subtracted from
    666 for files
    777 for directories
    to get the default permissions
  - umask default is 022, resulting in default permissions of
    rwxr-xr-x (755)

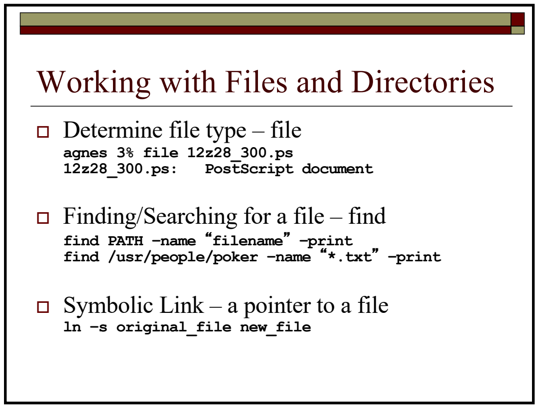
68



Working with Files and Directories

- □ Wildcard characters
  - * matches 0 or more of any characters
  - ? matches exactly one character
  - [Jj] matches exactly one J or j
  - [1-5] matches exactly one 1,2,3,4, or 5
  - ~ expands to full path to your home directory
  - ~poker expands to full path to poker’s home dir

69



Working with Files and Directories

- □ Determine file type – file

```
agnes 3% file 12z28_300.ps
12z28_300.ps:   PostScript document
```

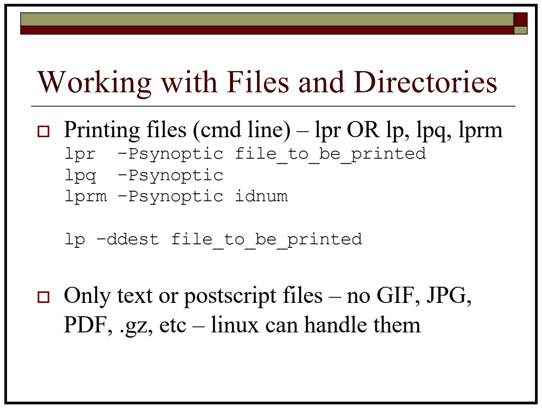
- □ Finding/Searching for a file – find

```
find PATH -name “filename” -print
find /usr/people/poker -name “*.txt” -print
```

- □ Symbolic Link – a pointer to a file

```
ln -s original_file new_file
```

70



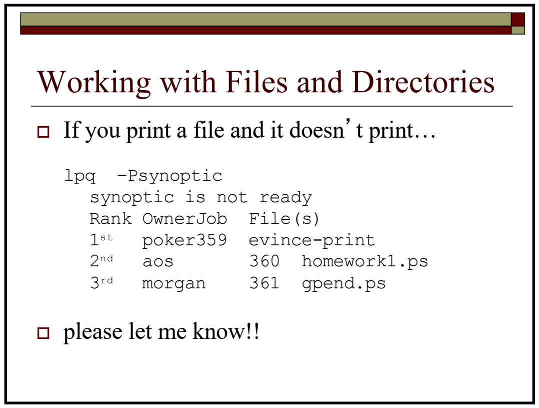
Working with Files and Directories

- □ Printing files (cmd line) – lpr OR lp, lpq, lprm

```
lpr  -Psynoptic file_to_be_printed
lpq  -Psynoptic
lprm -Psynoptic idnum

lp -ddest file_to_be_printed
```

- □ Only text or postscript files – no GIF, JPG, PDF, .gz, etc – linux can handle them

71



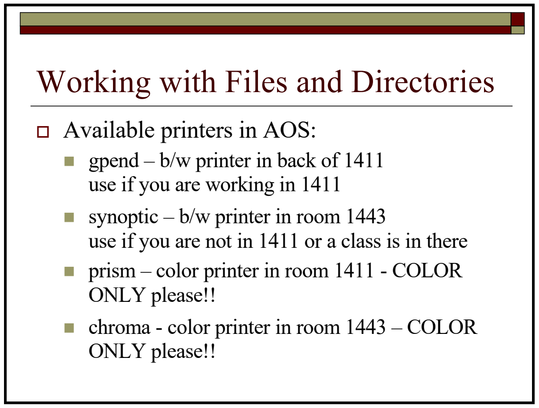
Working with Files and Directories

- □ If you print a file and it doesn’t print…

```
lpq  -Psynoptic
   synoptic is not ready
   Rank OwnerJob  File(s)
   1st  poker359  evince-print
   2nd  aos       360  homework1.ps
   3rd  morgan    361  gpend.ps
```

- □ please let me know!!

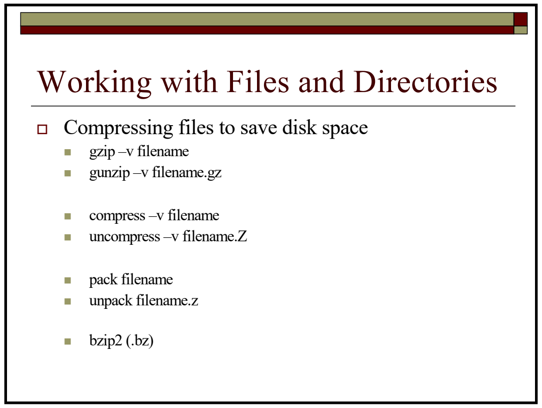
72



Working with Files and Directories

- Available printers in AOS:
  - gpend – b/w printer in back of 1411
    use if you are working in 1411
  - synoptic – b/w printer in room 1443
    use if you are not in 1411 or a class is in there
  - prism – color printer in room 1411 - COLOR ONLY please!!
  - chroma - color printer in room 1443 – COLOR ONLY please!!

73



Working with Files and Directories

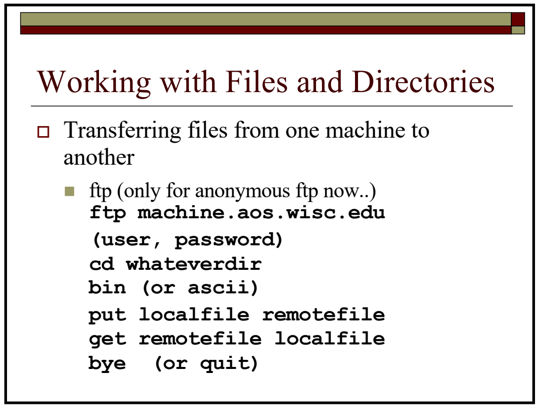
- Compressing files to save disk space
  - gzip –v filename
  - gunzip –v filename.gz

  - compress –v filename
  - uncompress –v filename.Z

  - pack filename
  - unpack filename.z

  - bzip2 (.bz)

74

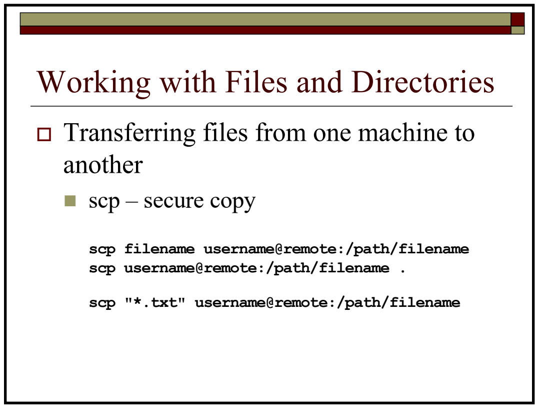


Working with Files and Directories

- Transferring files from one machine to another
  - ftp (only for anonymous ftp now..)

```
ftp machine.aos.wisc.edu
(user, password)
cd whateverdir
bin (or ascii)
put localfile remotefile
get remotefile localfile
bye  (or quit)
```

75

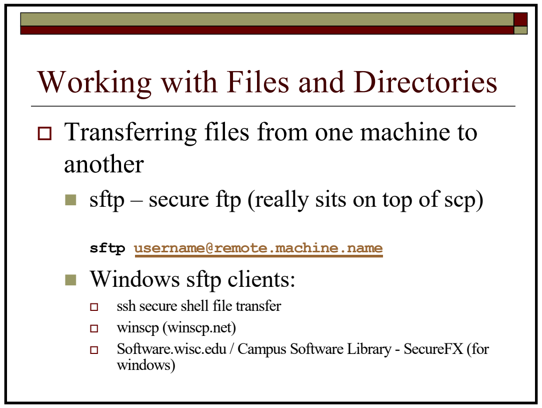


Working with Files and Directories

- Transferring files from one machine to another
  - scp – secure copy

```
scp filename username@remote:/path/filename
scp username@remote:/path/filename .

scp "*.txt" username@remote:/path/filename
```

76



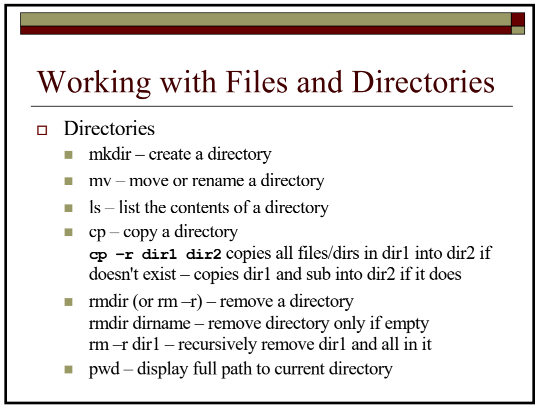
Working with Files and Directories

- Transferring files from one machine to another
  - sftp – secure ftp (really sits on top of scp)

```
sftp username@remote.machine.name
```

  - Windows sftp clients:
    - ssh secure shell file transfer
    - winscp (winscp.net)
    - Software.wisc.edu / Campus Software Library - SecureFX (for windows)

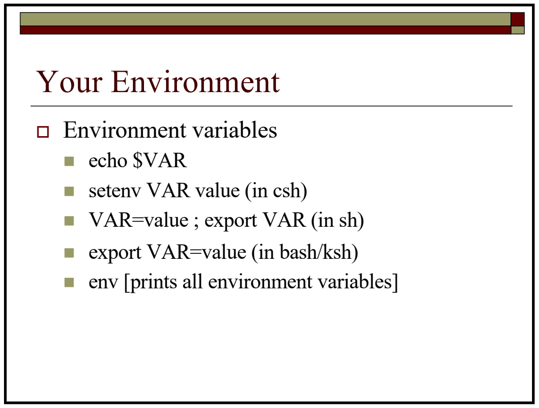
77



Working with Files and Directories

- Directories
  - mkdir – create a directory
  - mv – move or rename a directory
  - ls – list the contents of a directory
  - cp – copy a directory
    cp –r dir1 dir2 copies all files/dirs in dir1 into dir2 if doesn't exist – copies dir1 and sub into dir2 if it does
  - rmdir (or rm –r) – remove a directory
    rmdir dirname – remove directory only if empty
    rm –r dir1 – recursively remove dir1 and all in it
  - pwd – display full path to current directory

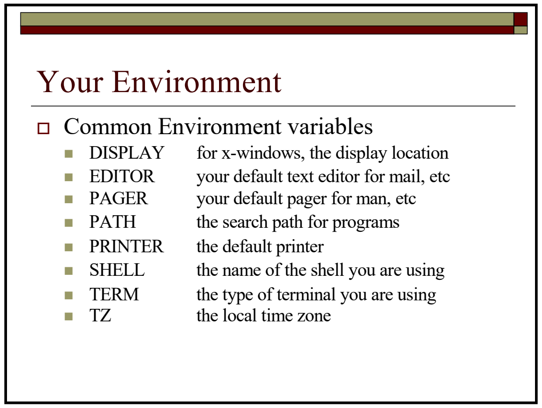
78



Your Environment

- Environment variables
  - echo $VAR
  - setenv VAR value (in csh)
  - VAR=value ; export VAR (in sh)
  - export VAR=value (in bash/ksh)
  - env [prints all environment variables]

79

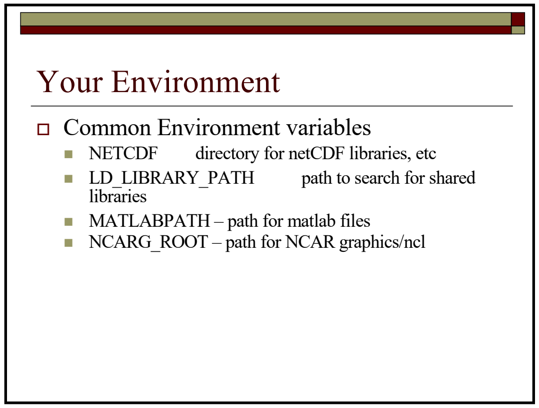


Your Environment

- Common Environment variables

DISPLAY	for x-windows, the display location
EDITOR	your default text editor for mail, etc
PAGER	your default pager for man, etc
PATH	the search path for programs
PRINTER	the default printer
SHELL	the name of the shell you are using
TERM	the type of terminal you are using
TZ	the local time zone

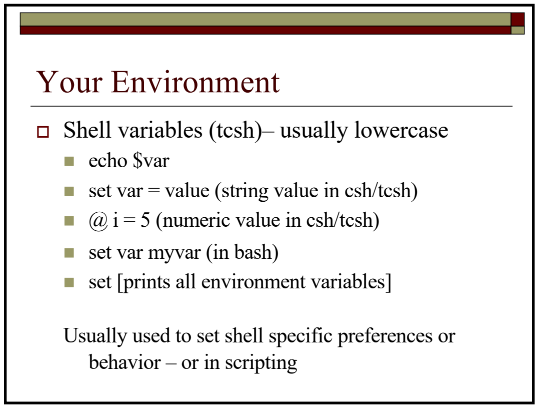
80



Your Environment

- Common Environment variables
  - NETCDF directory for netCDF libraries, etc
  - LD_LIBRARY_PATH path to search for shared libraries
  - MATLABPATH – path for matlab files
  - NCARG_ROOT – path for NCAR graphics/ncl

81

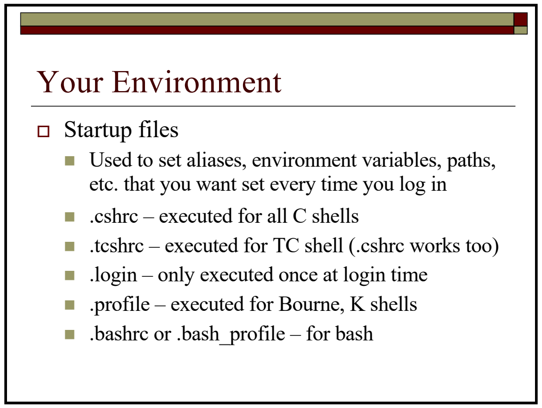


Your Environment

- Shell variables (tcsh)– usually lowercase
  - echo $var
  - set var = value (string value in csh/tcsh)
  - @ i = 5 (numeric value in csh/tcsh)
  - set var myvar (in bash)
  - set [prints all environment variables]

Usually used to set shell specific preferences or behavior – or in scripting

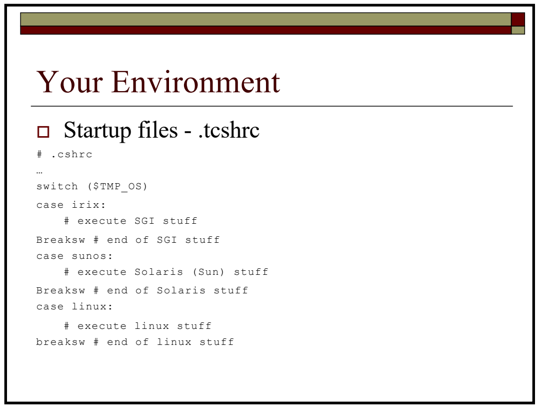
82



Your Environment

- Startup files
  - Used to set aliases, environment variables, paths, etc. that you want set every time you log in
  - .cshrc – executed for all C shells
  - .tcshrc – executed for TC shell (.cshrc works too)
  - .login – only executed once at login time
  - .profile – executed for Bourne, K shells
  - .bashrc or .bash_profile – for bash

83

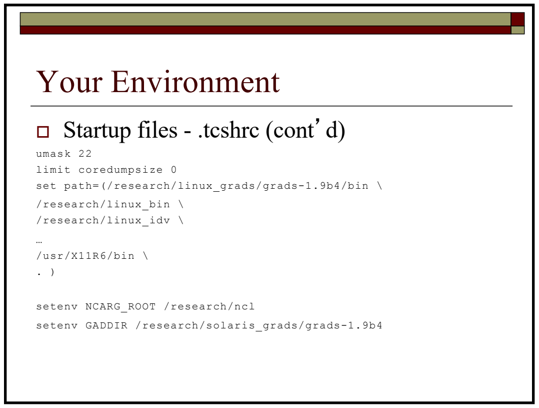


Your Environment

- Startup files - .tcshrc

```
# .cshrc
…
switch ($TMP_OS)
case irix:
    # execute SGI stuff
Breaksw # end of SGI stuff
case sunos:
    # execute Solaris (Sun) stuff
Breaksw # end of Solaris stuff
case linux:
    # execute linux stuff
breaksw # end of linux stuff
```

84

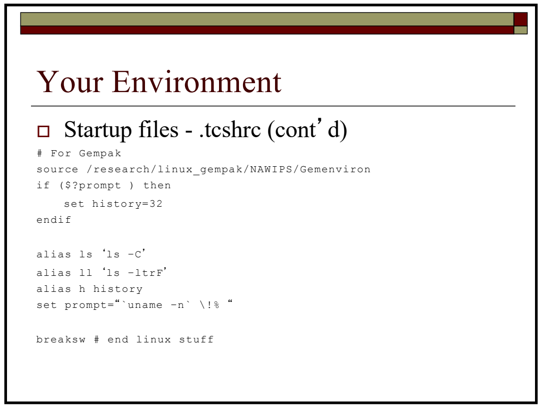


Your Environment

- Startup files - .tcshrc (cont’ d)

```
umask 22
limit coredumpsize 0
set path=(/research/linux_grads/grads-1.9b4/bin \
/research/linux_bin \
/research/linux_idv \
…
/usr/X11R6/bin \
. )

setenv NCARG_ROOT /research/ncl
setenv GADDIR /research/solaris_grads/grads-1.9b4
```

85

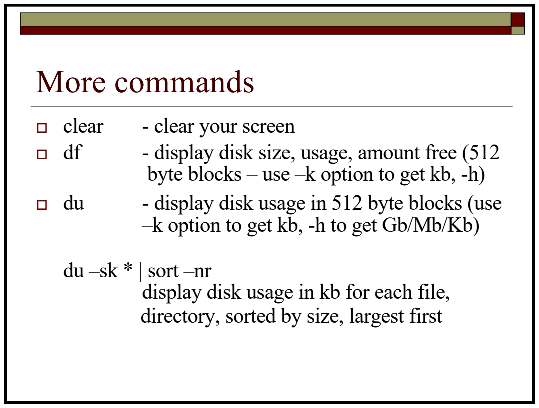


Your Environment

- Startup files - .tcshrc (cont’ d)

```
# For Gempak
source /research/linux_gempak/NAWIPS/Gemenviron
if ($?prompt ) then
    set history=32
endif

alias ls ‘ls -C’
alias ll ‘ls -ltrF’
alias h history
set prompt=“`uname -n` \!% “

breaksw # end linux stuff
```

86

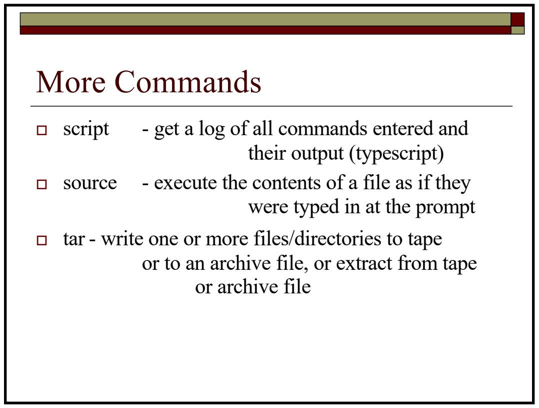


More commands

- clear - clear your screen
- df - display disk size, usage, amount free (512 byte blocks – use –k option to get kb, -h)
- du - display disk usage in 512 byte blocks (use –k option to get kb, -h to get Gb/Mb/Kb)

  du –sk * | sort –nr
  display disk usage in kb for each file, directory, sorted by size, largest first

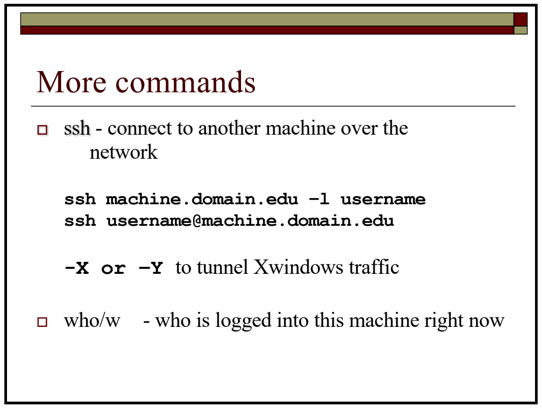
87



More Commands

- script - get a log of all commands entered and their output (typescript)
- source - execute the contents of a file as if they were typed in at the prompt
- tar - write one or more files/directories to tape or to an archive file, or extract from tape or archive file

88



More commands

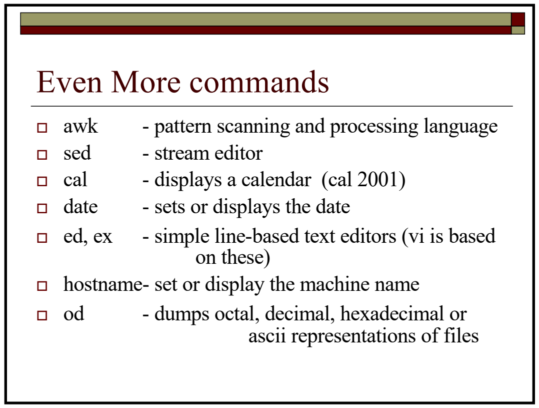
- ssh - connect to another machine over the network

  `ssh machine.domain.edu –l username`
  `ssh username@machine.domain.edu`

  `-X or –Y` to tunnel Xwindows traffic

- who/w - who is logged into this machine right now

89



Even More commands

- awk - pattern scanning and processing language
- sed - stream editor
- cal - displays a calendar (cal 2001)
- date - sets or displays the date
- ed, ex - simple line-based text editors (vi is based on these)
- hostname- set or display the machine name
- od - dumps octal, decimal, hexadecimal or ascii representations of files

90

Compilers/Programming languages

- cc/gcc C compiler
- CC/g++- C++ compiler
- f77/g77/pgf77/ifort - Fortran 77 compiler
- f90/g95/pgf90/ifort - Fortran 90/95 compiler
- gfortran Fortran compiler

  `gcc file.c` - produces a.out

  `gfortran –o exefile file.f` – creates exefile

91

Compilers/Programming languages

- perl Perl interpreter
- python python interpreter

```
python file.py
jupyter notebook (web IDE)
```

- javac/java Java compiler/interpreter
- make maintain, update, regenerate programs and files

```
make
make –f Makefile
```

92

Email

- mail – standard unix mail program

```
mail user@email.address
input text blah blah blah
.
```

- Mail – slightly more advanced
- elm
- Pine
- mozilla/thunderbird
- web email clients – in firefox/chrome

93

Text Processing

- Postscript – file begins with !PS…
  preview with ‘ggv, ghostview, gv’
  print using ‘lpr’
- tex/latex – dvi files – xdvi, dvipdf, dvips
- nroff/troff – old, mostly unix man pages
- xpdf or evince – read pdf files (or in chrome/firefox)

94

Image/movie processing

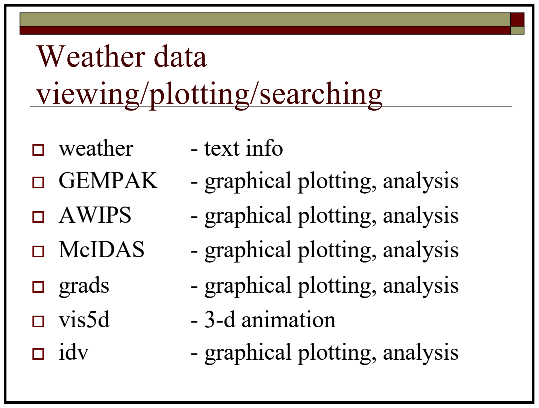
- pbmplus/netpbm- suite of image conversion progs
- ImageMagick - suite of image conversion progs (convert, display, identify, etc.)
- ffmpeg - movie creation/conversion
- xv - image viewer
- gimp - image prog similar to photoshop
- vlc - movie viewer
- xanim - animation/movie viewer

95

Web browsing

- NO MS Internet Explorer / Edge / Safari
- Mozilla firefox
- Google chrome
- lynx - text based web browser
- links - text based web browser

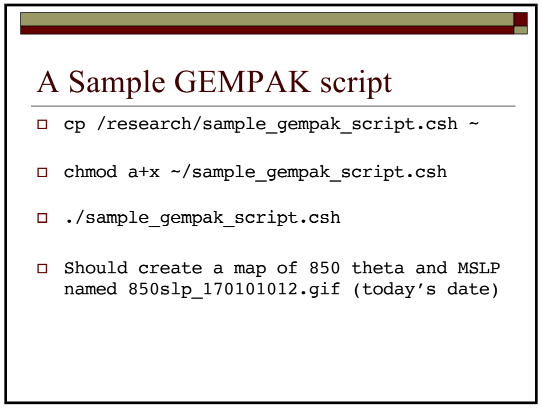
96



Weather data viewing/plotting/searching

- weather	- text info
- GEMPAK	- graphical plotting, analysis
- AWIPS	- graphical plotting, analysis
- McIDAS	- graphical plotting, analysis
- grads	- graphical plotting, analysis
- vis5d	- 3-d animation
- idv	- graphical plotting, analysis

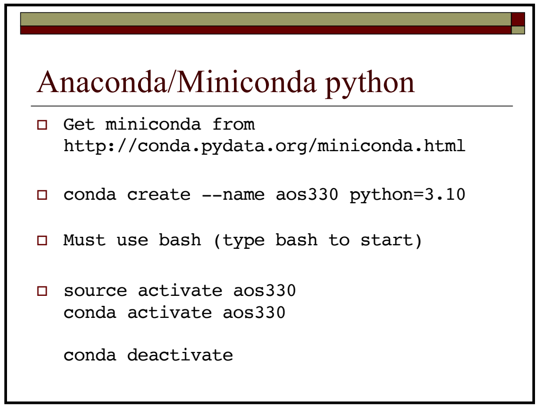
97



A Sample GEMPAK script

- `cp /research/sample_gempak_script.csh ~`
- `chmod a+x ~/sample_gempak_script.csh`
- `./sample_gempak_script.csh`
- `Should create a map of 850 theta and MSLP named 850slp_170101012.gif (today's date)`

98

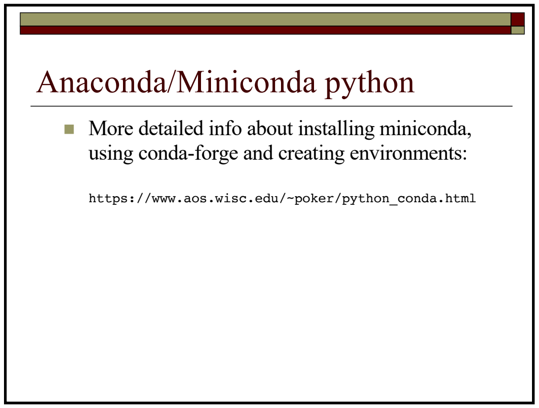


Anaconda/Miniconda python

- `Get miniconda from http://conda.pydata.org/miniconda.html`
- `conda create --name aos330 python=3.10`
- `Must use bash (type bash to start)`

-
  ```
  source activate aos330
  conda activate aos330

  conda deactivate
  ```

99

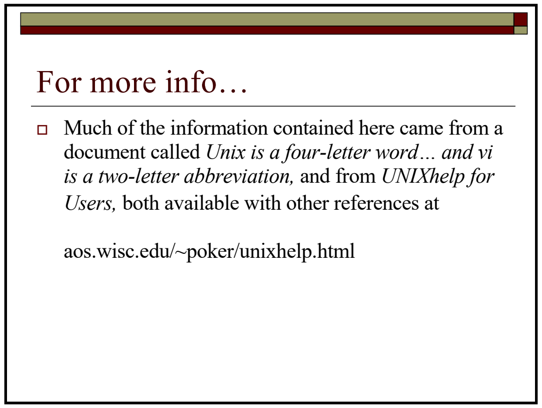


Anaconda/Miniconda python

- More detailed info about installing miniconda, using conda-forge and creating environments:

  `https://www.aos.wisc.edu/~poker/python_conda.html`

100

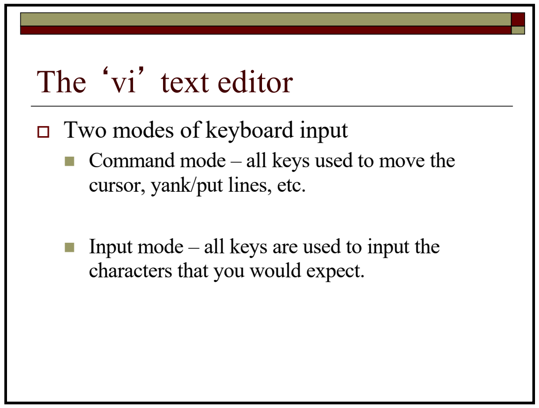


For more info…

- Much of the information contained here came from a document called *Unix is a four-letter word… and vi is a two-letter abbreviation,* and from *UNIXhelp for Users,* both available with other references at

  aos.wisc.edu/~poker/unixhelp.html

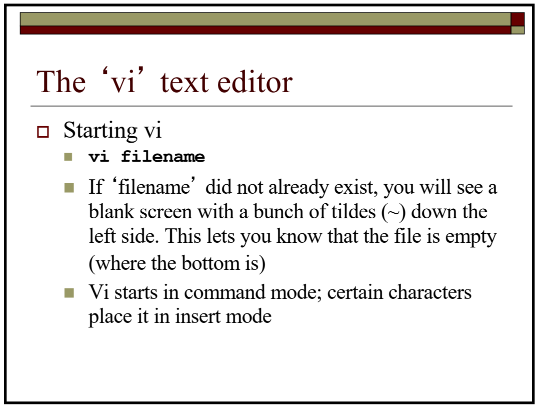
101



The 'vi' text editor

- Two modes of keyboard input
  - Command mode – all keys used to move the cursor, yank/put lines, etc.
  - Input mode – all keys are used to input the characters that you would expect.

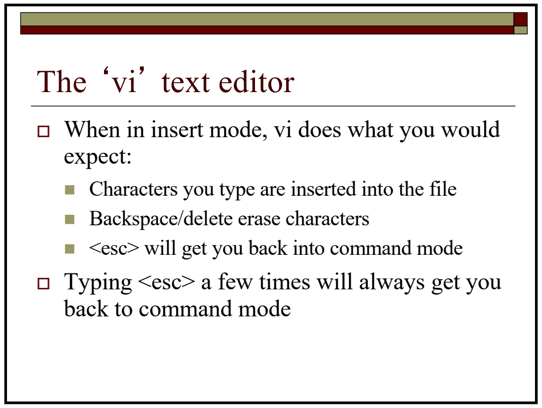
102



The ‘vi’ text editor

- Starting vi
  - `vi filename`
  - If ‘filename’ did not already exist, you will see a blank screen with a bunch of tildes (~) down the left side. This lets you know that the file is empty (where the bottom is)
  - Vi starts in command mode; certain characters place it in insert mode

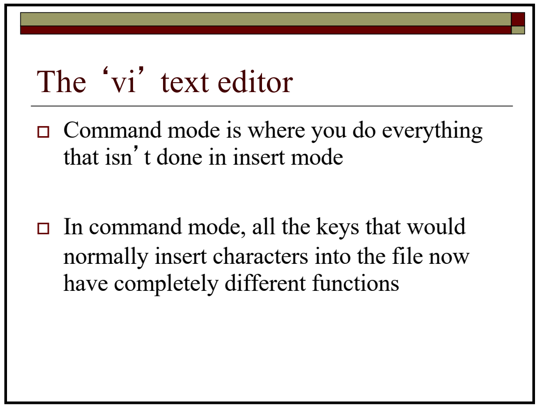
103



The ‘vi’ text editor

- When in insert mode, vi does what you would expect:
  - Characters you type are inserted into the file
  - Backspace/delete erase characters
  - <esc> will get you back into command mode
- Typing <esc> a few times will always get you back to command mode

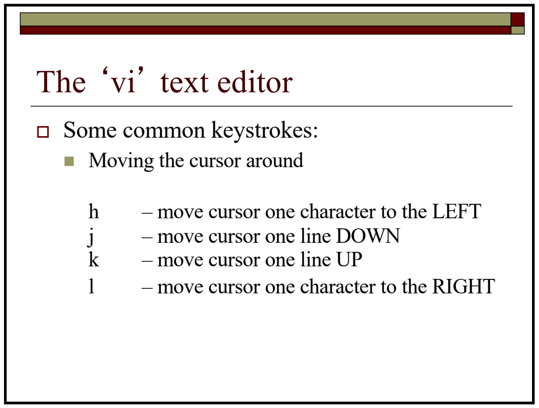
104



The ‘vi’ text editor

- Command mode is where you do everything that isn’t done in insert mode
- In command mode, all the keys that would normally insert characters into the file now have completely different functions

105

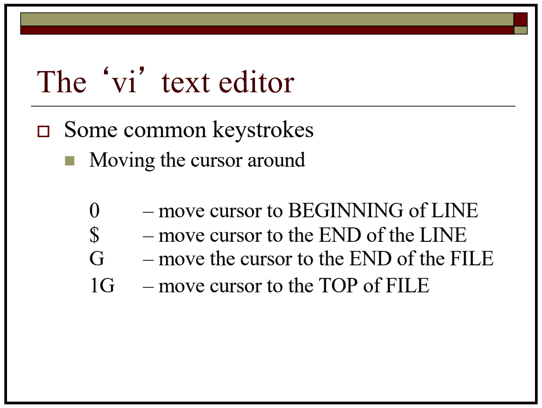


The ‘vi’ text editor

- Some common keystrokes:
  - Moving the cursor around

h	– move cursor one character to the LEFT
j	– move cursor one line DOWN
k	– move cursor one line UP
l	– move cursor one character to the RIGHT

106

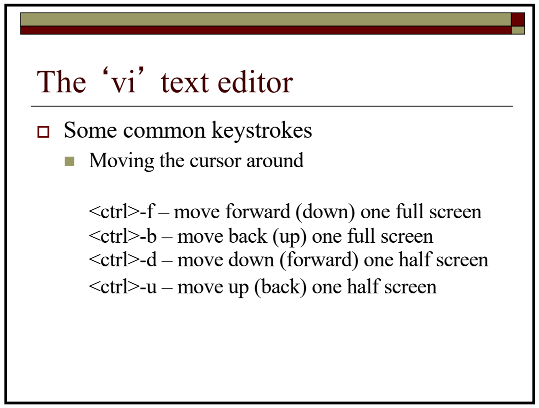


The ‘vi’ text editor

- Some common keystrokes
  - Moving the cursor around

0	– move cursor to BEGINNING of LINE
$	– move cursor to the END of the LINE
G	– move the cursor to the END of the FILE
1G	– move cursor to the TOP of FILE

107

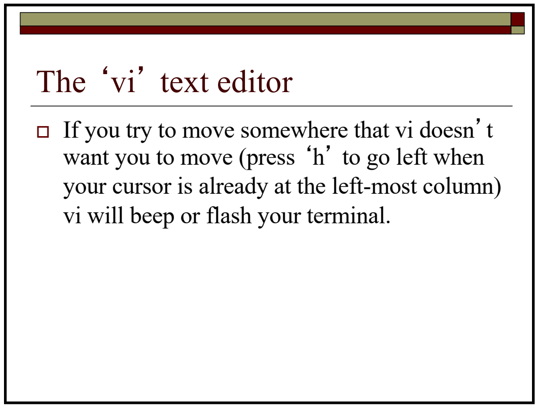


The ‘vi’ text editor

- Some common keystrokes
  - Moving the cursor around

<ctrl>-f – move forward (down) one full screen
<ctrl>-b – move back (up) one full screen
<ctrl>-d – move down (forward) one half screen
<ctrl>-u – move up (back) one half screen

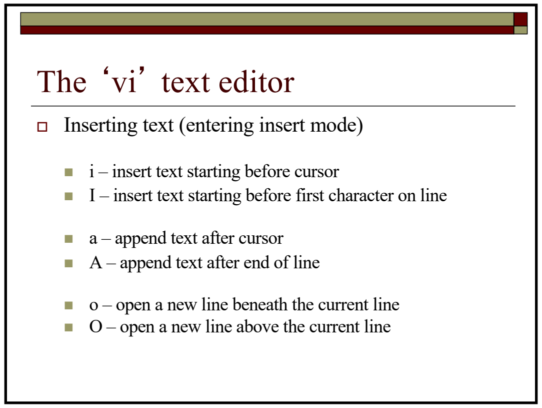
108



The ‘vi’ text editor

- If you try to move somewhere that vi doesn’t want you to move (press ‘h’ to go left when your cursor is already at the left-most column) vi will beep or flash your terminal.

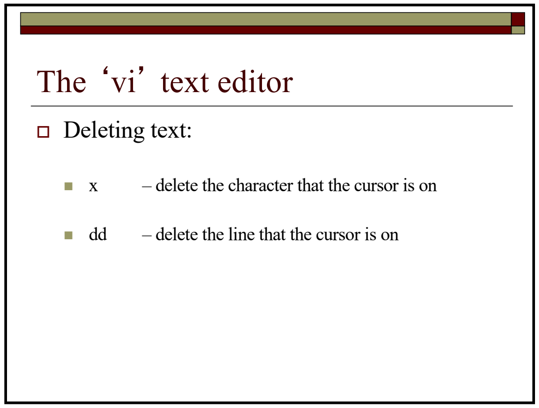
109



The ‘vi’ text editor

- Inserting text (entering insert mode)
  - i – insert text starting before cursor
  - I – insert text starting before first character on line
  - a – append text after cursor
  - A – append text after end of line
  - o – open a new line beneath the current line
  - O – open a new line above the current line

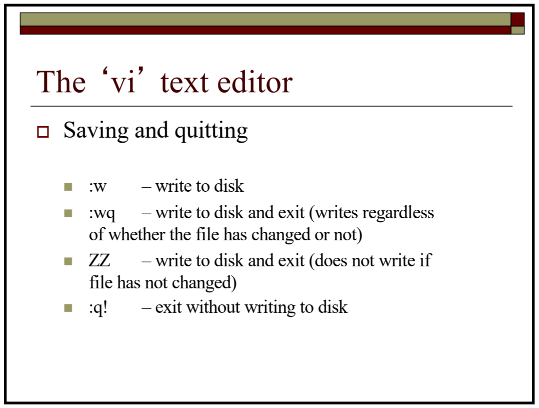
110



The ‘vi’ text editor

- Deleting text:
  - x – delete the character that the cursor is on
  - dd – delete the line that the cursor is on

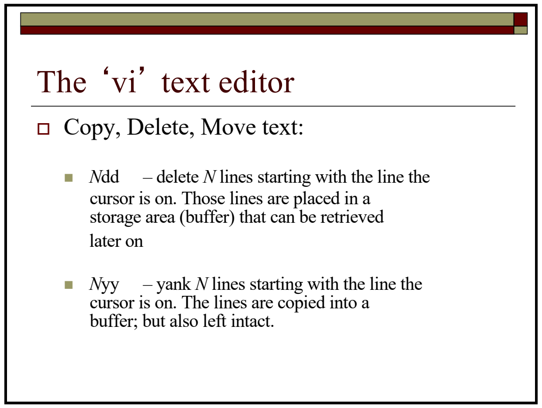
111



The ‘vi’ text editor

- Saving and quitting
  - :w – write to disk
  - :wq – write to disk and exit (writes regardless of whether the file has changed or not)
  - ZZ – write to disk and exit (does not write if file has not changed)
  - :q! – exit without writing to disk

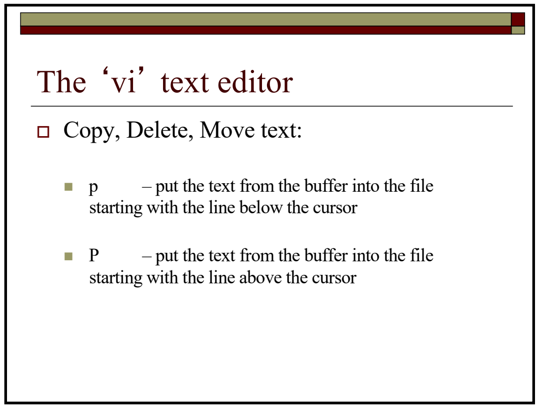
112



The ‘vi’ text editor

- Copy, Delete, Move text:
  - *N*dd – delete *N* lines starting with the line the cursor is on. Those lines are placed in a storage area (buffer) that can be retrieved later on
  - *N*yy – yank *N* lines starting with the line the cursor is on. The lines are copied into a buffer; but also left intact.

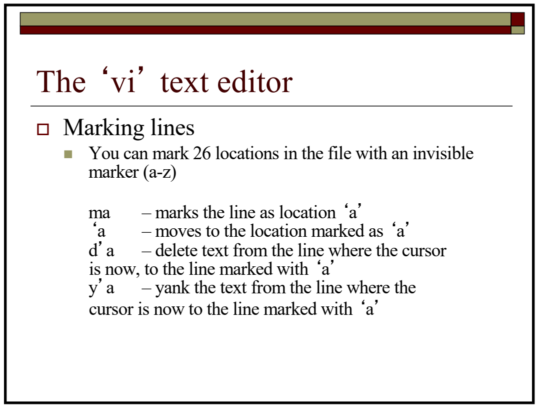
113



The ‘vi’ text editor

- Copy, Delete, Move text:
  - p – put the text from the buffer into the file starting with the line below the cursor
  - P – put the text from the buffer into the file starting with the line above the cursor

114

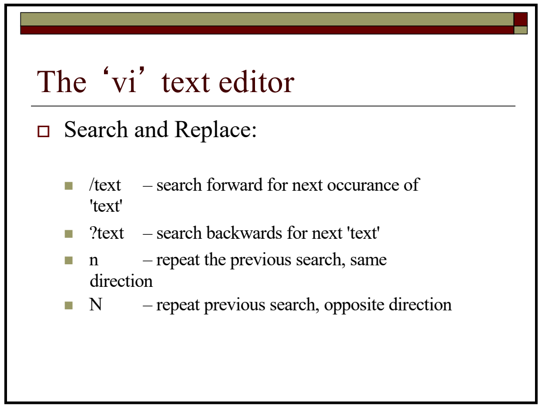


The ‘vi’ text editor

- Marking lines
  - You can mark 26 locations in the file with an invisible marker (a-z)

    ma – marks the line as location ‘a’
    ‘a – moves to the location marked as ‘a’
    d’a – delete text from the line where the cursor is now, to the line marked with ‘a’
    y’a – yank the text from the line where the cursor is now to the line marked with ‘a’

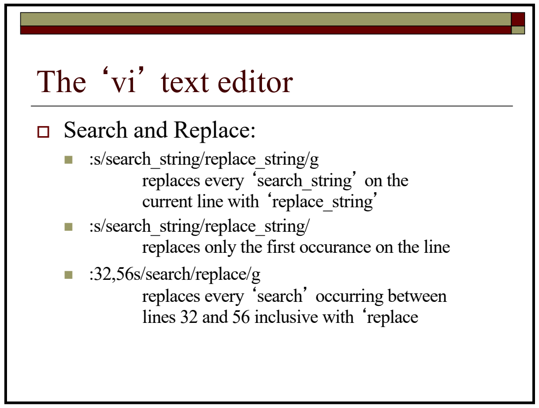
115



The ‘vi’ text editor

- Search and Replace:
  - /text – search forward for next occurance of 'text'
  - ?text – search backwards for next 'text'
  - n – repeat the previous search, same direction
  - N – repeat previous search, opposite direction

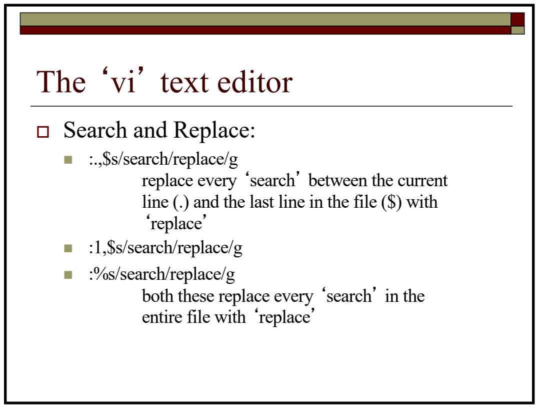
116



The ‘vi’ text editor

- Search and Replace:
  - :s/search_string/replace_string/g
    replaces every ‘search_string’ on the current line with ‘replace_string’
  - :s/search_string/replace_string/
    replaces only the first occurance on the line
  - :32,56s/search/replace/g
    replaces every ‘search’ occurring between lines 32 and 56 inclusive with ‘replace

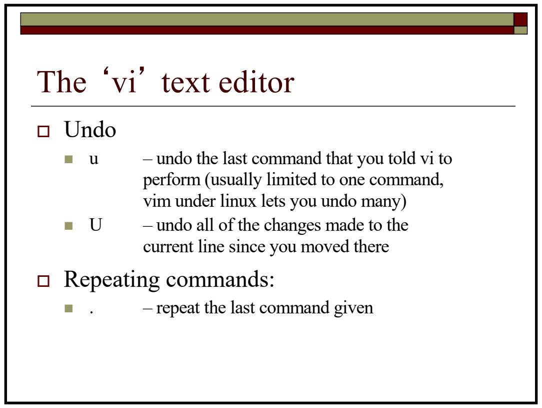
117



The ‘vi’ text editor

- Search and Replace:
  - :.,$s/search/replace/g
    replace every ‘search’ between the current line (.) and the last line in the file ($) with ‘replace’
  - :1,$s/search/replace/g
  - :%s/search/replace/g
    both these replace every ‘search’ in the entire file with ‘replace’

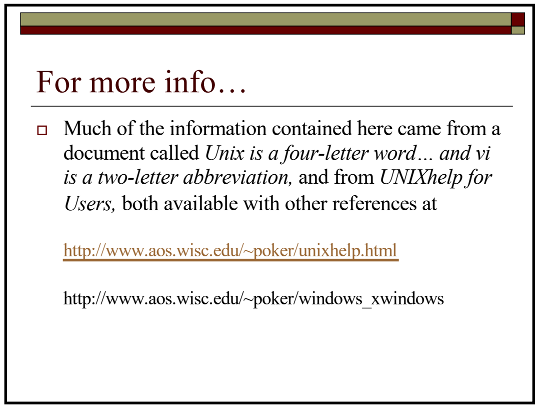
118



The ‘vi’ text editor

- Undo
  - u – undo the last command that you told vi to perform (usually limited to one command, vim under linux lets you undo many)
  - U – undo all of the changes made to the current line since you moved there
- Repeating commands:
  - . – repeat the last command given

119



For more info…

- Much of the information contained here came from a document called *Unix is a four-letter word… and vi is a two-letter abbreviation,* and from *UNIXhelp for Users,* both available with other references at

  http://www.aos.wisc.edu/~poker/unixhelp.html

  http://www.aos.wisc.edu/~poker/windows_xwindows

120