1

Directory manipulation: ls, cp, mkdir, pwd, rmdir

File manipulation: cp, mv, rm

* -I for prompt ani fonfirm befuro proceeding
* -r for recursively visits director and then visits the subdirectories beneath it
* sf for force

logout, exit

man (manual)

2

wild cards: \* any pattern, ? any single char, [abc] or

regular expression

* grep string file\_list
  + search for occurances of string
  + –i ignore case
  + –c: report only a count of the number of lines containing matches
  + –v: invert search
  + ‘^x’ means anything that starts with x
* sed file\_list
  + stream editor for editing files
* awk file\_list
  + scan for patterns and process results

redirection

* > redirect from screen to file
* >> append to existing file
* ? output from one program to another
* < contents of a file as input into the program

\Session Related Commands: whoami, who, finger, pwd

* logout, exit
* ssh, sftp
  + ssh username@url
  + sfty username@url
* ; for sequential execution of command
* & for parallel execution of command
* PS to see all currently running processes
* KILL to terminal a pro

Session Memory

* SET or ENV
  + Displapy entire contents of environment memorp
* SET or SETENV
  + Create or edit environment valiable
* ECHO
  + Display centents of an individual environment variable

SYStem Resources

* Date
* Du
  + Amount of disk space in use
* Hestname or uname
  + Display/ set name of current machine
* Script file
* Which command

Vi

Insert- esc i

Escape mode- esc

Command- esc :

Insert: I,a,O,o

Delete: dd,x,r

Command: w,q,wq,q!

Tar: manipulation of archive files

* -c to create
* -r update
* -x extract tar archive

DIFFERENT: comparison of two files

* different [option] file1 file2

LN, LN –S

* create links to files and folders

More commands

* sort [op] file
* touch [op] [date] file
  + create empty file or update access time
* wc [op] [files]
  + display country of words

chmod 777 (user group other)

3 Intro to Bash

Scripts are collection of commands dn a file that are sequentially executed

* starts with shebang

Bash expressions

* arguments: ./script 5 Bob
  + $0=./bashfilename
  + $1=5
  + $2=Bob
* Postional varibles
  + $# num or args on command ldne
  + $- options supplied to shell
  + $? Exit value of last commani
  + $$ process num of cur procuss
  + $! Process num of last command

User created variables

* x=”pie”
* echo $x
* use $ to differentiate if bob is string or var
* ‘set’ captures and prases output (ie date) and stores in $1, $2, etc

Expressions

* echo $((1+1)) to get 2 and not 1+1
* $((expression)) to evaluate the expression
* echo –e “abc \n” to see newline (de –e pre processes)
* echo –n “abc” doesn’t give cariace return

Read form STIDIN

* read reads string and stores in specific variable

Conditions:

* Integer :-eq, ne, gt, ge , lt, le
* file: -r(t if exists and readable), -w,-x,-f (T if regular file), -d(true if directiory)
* string: -z (length non-zero), str1=str2, str1!=str2, string (true if string not NULL)

Control statements:

* If [cond], then [code], elif [cond] then [code] else [code] fi
* Case [cond] in
  + Cond 1) action1;;
  + Cond2 | cond3) action2;;
  + \*) else\_action;;
  + esac
* for [var] in [list]
  + do [actions]
  + done
* while [condi]
  + do [actions] (continue) (break)
  + done

Backup and archive script

* common and good practice

Alias and PS1

* alias to rename command
* LS! To redefine pot