* Tuesday 11am—2pm in TA room
* Export PATH=$PATH: /usr/CS=/usr/CS:$PATH
* Sort, comm, and tr
* Sort: sort lines of text files
* Sort[options]…[FILE]: cat file | sort
* Comm[option]…FILE 1 FILE 2
* Comm FILE1 FILE2
* Output: lines unique to the first file lines only exist for the second file
* Lines both files have in common
* Tr[OPTION]…SET1[SET2]
* Cat file1/tr ‘[A-Z]’ ‘[a-z]’
* -d
* #!(shebang) /bin/bash : what kind of interpreter is needed
* rm … create a child shell (terminal)
* cp... executed by child shell
* mv… terminated and goes back to parent shell (terminal)
* extension: .sh
* sh is a symbolic link
* sh --> csh
* script
* #! /bin/bash
* rm –r(recursive) lab
* mkdir lab
* touch lab/lab.log
* touch lab/lab.txt
* touch lab/hw.txt
* ./ script
* (permission denied)
* chmod u+x script
* set –x: tell you what command is being excuted
* set +x: to turn it off (breakpoint???)
* add “-e” after echo option to activate “\n”
* printf ”%.3e\n” “.” and “e” means scientific, “3” decimal places
* variable using var=”hello”
* reference using echo $var
* local variables
* environment variable is global
* Exit: return value
* Check exit status of last command that ran with $?
* Echo $? Print the exit status to last command
* $reference a variable
* echo$
* print out the number of the arguments
* e.g. ./script ~ lauren
* output:2
* Accessing arguments
* Positional parameters represent a shell script’s command-line arguments
* For historical reasons enclose the number in braces if it is greater than 9
* 1: user name 2: passw
* user\_password =$2
* if statements use the test command or []
* man test to see the expressions that can be done
* -gt >

-lt<

-eq ==

neq !=

if []

then

elif[]

else

fi(end if)

example: if[$# -neq 2]

then echo”wrong # of argu”

exit 1

quotes

single quotes: do not expand at all, literal meaning

double quotes: expand backticks and $

Backtick ``or $

Expand as shell commands

Files=`ls -l` $(ls -l) executed and stored

Let count = count +1

./script read,blue,green

IFS=’,’

User\_input=$1

For color in @user-input

Do echo-$color

Done

Regular expressions

. match a single character

a\* nothing a aa aaa

a\*b nothingb ab aab aaab

^ has to be the beginning of the line

$ has to be at the end of the line

[a-d] match a, b, c, or d

grep: use basic regular expressions (BRE) : lose special meaning

cat file| grep s$

basic extended

regular \? \$ ? $

literal ? $ \? \$

egrep (or grep -E)

fgrep fixed string

cat file

sed ‘s(substitute)/Sunday/Saturday/[g](global, all the occurrence)’

* quantification: {n} a{3,6} inclusive three to six times
* Grouping: a(bc)\*
* Alternation: [x,y,z] single character (boy|girl)
* Replace all words in <> in a file, with just words
* Cat file| sed ‘s/<\(.\*\)>/ \ 1(/g, for all the cases’
* Cat file | sed ‘s/<\(.\*\)>,<\.\*\>/\1,\2/g

Find all lines in a file that start and end with the same word

Cat test2 | grep “^\([A-Za-z]\+\).\*\1$”