Week 2
Shell Scripting, RegEx, and Streams

10 October 2018

CS 35L Lab 4

Jeremy Rotman

Announcements

- Assignment #1 **was** due October 6 by 11:55pm
  - You can still submit the assignment
  - If you submit before 11:55pm tonight, it's only an 8% penalty!
- Assignment #2 is due October 17 (Wednesday) by 11:55pm
- For Assignment #10
  - Feel free to begin choosing articles
  - Email me to tell me what you are choosing
  - I will set up a sheet listing who is doing which article so you can see what is taken already
  - Also look into the resources for written and oral presentations

Questions?

Outline

- ➔ Useful shortcuts
- ➔ RegEx examples
- ➔ Homework 2 Tips

Some useful shortcuts

- `ctrl-c`
  - Cancel any running command
  - Useful if you're running something but realize that it won't actually work
- `ctrl-z`
  - Sends the current process to the background
  - Useful to leave emacs without actually exiting
  - `fg` returns the last program you pushed to background to the foreground

Some useful shortcuts

- `ctrl-l`
  - Clears your terminal screen
  - Essentially just pushes everything above what you can see
  - Useful to make your screen a bit less of a mess
  - Alternatively use the command clear
- `ctrl-u`
  - Deletes the typed line behind the cursor
  - Useful if you've written a large command you no longer want to run

Some useful shortcuts

- `ctrl-a`
  - Moves cursor to the beginning of the line
  - Useful if you forgot to type in the actual command
- `ctrl-e`
  - Moves the cursor to the end of the line
  - Useful if you made an edit but want to now continue typing

RegExamples

Assume for the following that we are using egrep (grep -E) to search for lines in a file

RegExamples

What would you use to match lines that begin with a “th” not case specific?

RegExamples

What would you use to match lines that begin with a “th” not case specific?

^[Tt][Hh]

RegExamples

How about lines that end with punctuation?

RegExamples

How about lines that end with punctuation?

[[:punct:]]$

RegExamples

What would the following RegEx give you?

[[:digit:]][ ]+[[:digit:]]

RegExamples

What would the following RegEx give you?

[[:digit:]][ ]+[[:digit:]]

Any line where there are two digits separated by 1 or more spaces

RegExamples

Given the RegEx: ((do*t) \2)+

Which of the following would match?

A. doot doot
B. doot doot doot
C. doot
D. doot doot Mr. Skeltal

RegExamples

Given the RegEx: ((do*t) \2)+

Which of the following would match?

A. doot doot
B. doot doot doot
C. doot
D. doot doot Mr. Skeltal

RegExamples

Given the RegEx: [Hh]ello.*[Ww]orld

Which of the following would match?

A. Hello World
B. helloworld
C. Hello to the most beautiful world I have ever seen
D. Hi World

RegExamples

Given the RegEx: [Hh]ello.*[Ww]orld

Which of the following would match?

A. Hello World
B. helloworld
C. Hello to the most beautiful world I have ever seen
D. Hi World

RegExamples

Given the Line: LEEEEEEEEEEROY JENKINS

Which of the following regular expressions would match this?

A. LE{3}ROY JENKINS
B. LE{2,30}ROY JENKINS
C. LE?ROY JENKINS
D. L[a-z]*ROY JENKINS

RegExamples

Given the Line: LEEEEEEEEEEROY JENKINS

Which of the following regular expressions would match this?

A. LE{3}ROY JENKINS
B. LE{2,30}ROY JENKINS
C. LE?ROY JENKINS
D. L[a-z]*ROY JENKINS

Homework #2

- Search through a directory for files that are duplicated
  - Replace duplicates with hard links to a single copy of the file
    - It should keep the file that is lexicographically first
    - Initial preference to files beginning with “.”
- Your script should take a single argument D
  - This is the directory your script should search
  - It should only search the directory
    - Do not recursively examine subdirectories
- Your script should ignore symbolic links and directories
- Be prepared to handle files with special characters in their names (e.g. space, *, or a leading -)

Homework #2

- ➔ You do not need to worry about being given 0 arguments or more than 1 argument
- ➔ If your program encounters an error reading a file, report the error, but do not count it as a duplicate file
- ➔ Your script should be runnable as an ordinary user

Homework #2 Hints

- `cmp` *`file1 file2`*
  - Compares two files
  - Write no output if they are the same
- `ln` *`source_file target_file`*
  - Creates a link to the source file at the target file
  - -f option forces existing destination pathnames to be removed to allow the link

Homework #2 Hints

- `test [expression]`
  - -L
    - True if the pathname resolves to an existing directory entry for a symbolic link
  - -f
    - True if the pathname resolves to an existing directory entry for a regular file
- `ls -i`
  - Displays the inode numbers
    - An inode is a data structure that stores information about files
    - Files that are hard-linked will share the same inode number

Bash Arrays

- `declare -a hw`
  - Creates an array named hw
- `hw[0]="hello"`
  - Sets the first (zero-based indexing) element of hw to “hello”
- `for x in ${hw[@]}`
  - Iterates over all elements of hw
  - @ references all members of the array
- `${hw[@]}`
  - References the array

Bash Arrays

- `${#hw[@]}`
  - Length of the array
- `${#hw[0]}`
  - Length of element 0 in the array