CS35L – Fall 2018

Slide set:	2.2
Slide topics:	Shell scripting, regex, streams
Assignment:	2

Regular Expression Review

4 Basic Concepts

- Quantification
  - How many times of previous expression?
  - Most common quantifiers: ?(0 or 1), *(0 or more), +(1 or more)
- Grouping
  - Which subset of previous expression?
  - Grouping operator: ()
- Alternation
  - Which choices?
  - Operators: [] and |
    - Hello|World [A B C]
- Anchors
  - Where?
  - Characters: ^ (beginning) and $ (end)

RegEx Exercises

- Which of the following strings would match the regular expression: aab?b
  - A. aabb
  - B. aa\nbbb
  - C. aab

Answer: aabb

aab

RegEx Exercises

- Which regular expression would match the words “favorite” and “favourite”?
  - Answer: “favou?rite”

RegEx Exercises

- Which regular expression would match the words “Ggle”, “Gogle” and “Google”?
  - Answer: “Go*gle”
- Which one would match “Gogle”, “Google” and “Gooogle” but not “Ggle”?
  - Answer: “Go+gle”

RegEx Exercises

- Which regular expression would match any version of the word “Google” that has an even number of o’s?
  - Answer: “G(oo)+gle”

- Which regular expression would match any version of the word “Google” that has fewer than 7 O’s?
  - Answer: “Go{0,6}gle”

RegEx Exercises

- Which line(s) would this regular expression match? “^T.+e$”
  - A. The
  - B. Te
  - C. Three
  - D. Then
  - E. The Two

  Answer: The, Three (ERE)

RegEx Exercises

- Which regular expression(s) would match the words “Ted”, “Ned” and “Sed”?
  - A. (T|N|S)ed
  - B. [T N S]ed
  - C. .ed
  - D. [L-U]?ed
  - E. .*ed

  Answer: A., B., C.,
  D., E. (ERE)

RegEx Exercises

- Which regular expression would match all subdirectories within a directory?

  – Answer: ls –l | grep “^d”

Lab 2

Assignment 2 Details

- Submit 3 files:
  - Script “buildwords”
  - Simple text file “lab2.log”
    - 80 character limit per row
  - Script “sameln”
- Check everything on SEASnet!
  - Assignments graded on SEASnet servers (eg. lnxsrv07)

What is Lab 2 About?

Build a spelling checker for the Hawaiian language
(Get familiar with sort, comm and tr commands!)

- Steps:
  1. Download a copy of web page containing basic English-to-Hawaiian dictionary
  2. Extract only the Hawaiian words from the web page to build a simple Hawaiian dictionary. Save it to a file called hwords (**site scraping**)
  3. Automate site scraping: `buildwords` script (cat hwnwdseng.htm | buildwords > hwords)
  4. Modify the command in the lab assignment to act as a spelling checker for Hawaiian
  5. Use your spelling checker to check hwords and the lab web page for spelling mistakes

Useful Text Processing Tools

- wc: outputs a one-line report of lines, words, and bytes
- head: extract top of files
- tail: extracts bottom of files
- tr: translate or delete characters
- grep: print lines matching a pattern
- sort: sort lines of text files
- sed: filtering and transforming text

Lab2.log

- .log is the same as .txt – no difference
- Ex:
  - 1. I used wget to download the webpage
  - 2. I ….
  - 3. Answer to #3 here
- Should read basically like a lab journal
- Keep things concise!

Lab Hints

- Run your script on seasnet servers before submitting to CCLE
- sed '/patternstart/,/patternstop/d‘
  - delete patternstart to patternstop, works across multiple lines
    will delete all lines starting with patternstart to patternstop
- The Hawaiian words html page uses \r and \n for new lines
  - od –c hwnwdseng.htm   to see the ASCII characters
- You can delete blank white spaces such as tab or space using
  - tr -d '[:blank:]’
  - Use tr -s to squeeze multiple new lines into one
- sed 's/<[^>]*>//g' a.html to remove all HTML tags

Buildwords

- Hawaiian.html -> buildwords -> hwords
- Buildwords
  - Read from STDIN and perform work on input
  - Output to STDOUT
- Ex:
  - $ ./buildwords < hawaiian.html > hwords

Homework 2

- Write a script `sameln` that does the following:
  - User provides a directory name as an argument
  - Finds all regular files in directory(do not recurse) and ignores all other types (directories, symlinks, etc.)
  - If 2 or more files have the same content (cmp)
    - Keep the file whose name is alphabetically first OR starts with a dot
    - Replace duplicates with hard links (ln)
    - File names may contain special characters!
  - Hint: see the cmp, ln, and test utilities.

Checking Hard Links

- Inode: data structure that stores information about files
  - File type
  - Permission
  - Owner
  - File Size, etc.
- Each inode is identified by a unique inode number within the file system
- Check a file’s inode number: ls –i filename
- **How do you check if two files are hard-linked?**
  - They have the same inode number