Library Carpentry

Software skills for librarians

Module 1: The Unix shell
Exercises

Computational thinking

1. Imagine you have been asked to help send a series of information packs out. Each pack consists of one copy each of a number of sheets, to be sealed in an envelope and labelled with an address. The address labels have been ready printed. How would you undertake this task to perform it as quickly as possible?

   a. Identify the various stages involved in the task?
   b. What sequence should these be performed in?
   c. How would you arrange the materials to facilitate this?
   d. How would the task change if you had an additional person to help?

2. Suppose you are offering a comb binding service for student projects. Several students wish to use the service an hour before their projects are due.

   a. What are the stages in binding a project?
   b. Make an estimate about how long it takes to bind a project?
   c. Suppose the functions of punching a document and binding it are split across two machines. Does this help?
   d. What if you had two binding machines (each can punch and bind)?

Unix shell

3. Ensure that you can access the shell on your laptop.

4. Create a new directory to hold the example files.

5. Download the example files for this course, and unpack them to the directory you have just created.

6. List the contents of this directory; how many files are there? How much space do they occupy?

7. Practice moving about the directory tree. How do you get back to your home directory?

8 a. Using a text editor create a short file with a few lines of text. Try making a copy of this in another directory; try renaming it.
  b. From the command line display the contents of this file (without loading it back into an editor).

9 a. What are the disadvantages of this approach with a larger file such as gulliver.txt (supplied with the example files)
  b. What approaches could you take to display the file in more managable chunks?
  c. How can you find out how large the file is: as a number of words? Or as lines?

10. The marctxt script converts a file in MARC exchange format to its textual representation. There are three MARC records supplied in the example files. (invoke the script using `./marctxt <filename>`)

    a. Write a simple loop to convert all of these records to text.
    b. This time save the output to another file.

11. Use `grep` to find:

    a. All LCSH subject headings in these records.
    b. Those records coded as rda in the 040 field
    c. All RDA content, media and carrier type fields.
    d. All occurances of the publisher's name "Addison Wesley"
    e. If the word Linux only appears in the title?

12. This exercise will guide you through working on the gulliver.txt file. The object is to clean the file up and count the frequency of individual words.

    a. Use `sed` to remove the footer (lines 9352 to 9714).
    b. Similarly, use `sed` to remove the header (lines 1 to 37)
    c. Use the `tr` command to delete all punctuation marks. (Hint: you will need to use input redirection, and the class `[:punct:]` means all punctuation marks)
    d. Now use `tr` to convert all uppercase letters to their lowercase equivalent.
    e. Again use `tr` to replace all spaces with a newline.
    f. Sort the list of words into order.
    g. Now use `uniq -c` <filename> to remove the duplicates and produce a word count.

Regular expressions

13 a. What will Fr[ea]nc[eh] match?
   b. How about ^Fr[ea]nc[eh]$?
   c. How would you match strings beginning with only French and France?
   d. How would you match Colour and Color?

14. Write a regular expression to find any four letter word which ends a string.

15 a. How could you match dates in the format dd/mm/yyyy ?
   b. How about dates like 28 Feb 2017 ?

16 a. Write a regular expression to match 10 digit ISBN numbers without hyphens?
   b. Do the same for 13 digit ones (you may assume that these will all begin with 978).
   c. How about 10 digit ISBN numbers with hypens (you can accept either false positives or negatives)?

17 a. Match a three letter MARC language code (like eng, fre, ger).
   b. Match the language code preceeded by any one of subfields $a to $h.
   c. Match a complete 041 field consisting of one or more subfields, each with a single language code.