Assignment 3 Hints

CS 35L – Spring 2020

Don’t forget: **export LC_ALL=‘C’**

Submit all scripts with the exact names as specified on the assignment webpage!

Do not compress your files (zip, tarball, …)!

Lab Hints

- Make sure *buildwords* takes input from **stdin** and outputs to **stdout**

- Do not include file input redirection or file output redirection (i.e. **> words** or **< inputFile**)!

- Note, some English words consist entirely of Hawaiian letters (e.g. *pineapple*, *man*, *help*). To keep things simple, treat these words as Hawaiian words that the script can extract.

Lab Hints

- Follow these steps in *buildwords* (one long chain of piped commands):

- Use **sed** to remove ‘?’, ‘<u>’, ‘</u>’
- Use **tr** to map
  - ‘`’ (backtick) → ' (single quote)
  - ‘-’ (dash) → ‘ ‘ (space)
- Use **grep** to extract the lines of the form ‘*A*<td*X*>*W*</td>*Z’* (as described on the assignment webpage)
- Use **sed** to remove the ‘A<tdX>’ and the ‘</td>Z’ parts from the lines
- Use **tr** to squeeze and translate ‘ ‘ (spaces) into ‘\n’ (newlines)
- Use **tr** to map all uppercase letters to lowercase letters
- Use **sort** to sort the lines (retaining unique lines)

HW Hints

- The first thing you should work out is the regular expression to match the filenames that violate the specified guidelines due to invalid characters (not including the duplicates guideline).

  - One recommended way to do this is to write a regular expression that matches all the **valid** filenames (i.e. that do not violate the guidelines) and then use the *grep –v* option to select all the lines that do not match this regular expression.

  - One regular expression template to use:
    - '/…otherRegexHere…$'   (matching with the filename's last component)

HW Hints

- Duplicate filenames can have invalid characters too (e.g. *ans1.txt*, *Ans1.txt*). Only print an filename that violates the guidelines once! A suggested strategy:
  - Print all the filenames that have invalid characters first, then print all the filenames that have duplicates from the set of **valid** filenames (no need to print filenames with invalid characters twice. As an example, *ans1.txt*, *Ans1.txt* will each be printed once due to invalid characters. Do not print them again because they are duplicates).

HW Hints

- How to output the matched filenames in the proper format?

- Use **xargs** and **ls**

- Lookup the **xargs** and **ls** options that do the following:
    - Do not run **ls** if the standard input is empty
    - Use the newline control character ‘\n’ as a delimiter when parsing the input
    - Print each filename on a new line
    - Do not list the contents of directories
    - Append a forward slash ‘/’ to directories
    - Print the raw entries (do not treat control characters specially)
    - Show the control characters

HW Hints

- Avoid variable expansion or command substitution as arguments to commands.
- As an example, avoid:
  - **ls “${myfiles}”**
  - **ls “$(find /path/to/dir)”**

- Why? Bash will treat certain characters as special when they are expanded/substituted. Invalid filenames with characters such as ‘!’, ‘*’, ‘$’ will produce unwanted behavior in your script.
- Solution: Use **xargs** and **ls**

HW Hints

- How to deal with duplicate filenames?

- Use **sort** and **uniq** (remember to apply the suggested strategy described on slide 3).

- Lookup the **sort** and **uniq** options that do the following:
  - Sort while ignoring case
  - Ignore differences in case when comparing adjacent lines
  - Print all duplicate line groups

HW Hints

- How to capture all immediate directory entries?

- Use **find**

- Lookup the **find** options that implement the following:
  - Descend only to a level of 1 (do not enter subdirectories)
  - Avoid listing ‘.’ and ‘..’ (use the -*mindepth* option)

HW Hints

- How to implement recursion?

- Use **find**
  - Find all the directories under the given start directory *D* (including the start directory *D* itself). Call *poornames* with no recursive option on each directory (lookup the **find** option, -**exec**).
  - *poornames* with no recursion should list all the invalid filenames immediately under the passed start directory.

HW Hints

- Suggested *poornames* template:

```
#! /bin/bash
#------if statements and case statements---------
        #this block of code should check for:
                # -r passed? (use shift command here)
                #was a start directory passed as an argument?
                #is the start directory valid? If not, print to stderr and exit
                #wrong usage or wrong number of arguments? -> print to stderr and exit
#---------------recursion check------------------
        #this block of code checks for recursion:
                #if recursion -> run recursion statement
                #else -> run non-recursive block
#---------------non recursive block------------
        #this block of code could be a function
        #get all immediate filenames and print the ones that have invalid characters
        #find all immediate and valid filenames and print the duplicates
```