3.3bash-string-ops-lab

CST 334 (Operating Systems)
Dr. Glenn Bruns

# Lab: Bash string operations

We are going to use bash, awk, and make on a few problems. Please use mlc104 for this lab.

1. The directory /home/CLASSES/brunsglenn/cst334/labs/strings-lab contains some student files (the contents have been deleted).

   a. Write a bash script ‘get_ids.sh’ that will loop over the files in this directory, and for each file will display the student ID that is embedded in the file name. Your script should produce output like this:

   ```
   649584

   649579

   649545
   ```

   b. Write another script 'get_names.sh' that will display the student name associated with each file.

   c. Write another script 'get_submitted_name.sh' that will write the name that the student provided as the submitted file name (this is everything after ‘assignsubmission_file_’).

2. If you still have time, create a bash script to count the number of files in a directory (and below) by size category.

   a. use the ‘find’ command to find the names and sizes of all files at or below a certain directory -- use your home directory on mlc104 for the time being.

   b. put your find command into a file, and call it ‘find-sizes.sh’. For now your home directory can be “hard-wired” directly in this script. Don’t forget the shebang line.

   c. run your script and make sure it works properly

   hint: at this point, you can check your work by looking at the code in

   ```
   /home/CLASSES/brunsglenn/spring18/labs/find-lab/step-c
   ```

   and similarly for steps e, g, h, and k. (*This special answer-checking service has been made available through generous gifts from students like you.*)

   d. modify your script so that it takes a single command line parameter that specifies the directory that should be used

   e. modify your script so that it checks to make sure that a command-line parameter has been provided

   f. modify your script, using awk, so that only the file sizes (they must be in bytes) are output.

   g. write an awk script, named aggr.awk, that will take as input a file of positive integers, and will produce (to standard output) one integer for each integer, as follows: if the input integer is less or equal to 256, output 256. Else if the input integer is less than or equal to 512, output 512. Else if the input integer is less than or equal to 1024, output 1024. Etc.

   h. create a make file (called ‘Makefile’). Put an entry in the make file that will call your bash script and pipe the result to your awk script

i. test that everything is working

j. now, using sort and uniq, extend what is in your makefile so that the output will be the counts of different file sizes

k. make sure the result is working fine when your home directory is used as the directory specified to your bash script

l. When you are sure your script is working right, run your command on /home/CLASSES, and save the result.