Assignment 1 — is_same

Dr. William Kreahling

February 21, 2021

Objectives:

Learn to create a shell script, change its permissions so you can execute it, create comments within a shell script, access parameters on the command line, test for various conditions, use the `if` statements, use the `echo` command, use the `cmp` command, invoke a Linux tool from within script, output redirect from within a script,and exit a script with the correct exit value.

Instructions:

Your assignment is to write a *Bourne Again SHell*(bash) script that deletes a file (if appropriate). The shell script **must** be called `is_same.sh` For every deletion that occurs append a message indicating which what happened in a log file. The log file, named *log.txt* should contain a record of every attempted or successful deletion that has (or has not) occurred. *man the tool 'cmp' for help accomplishing your goal*

The requirements listed directly below will only be stated in this assignment but will be required in all subsequent shell scripts assigned this semester. In all of your shell scripts be sure to:

1. Put the *shebang* at the beginning of the script to indicate you are using the *Bourne Again SHell*.
2. Put comments after the first line to identify yourself, the assignment, and describe the general purpose of the script.
3. Put an appropriate comment before each block of commands in the script.
4. Exit with a status of 0 when the script has correctly accomplished its task.
5. Print an appropriate error message and exit with a non-zero status when the script was not able to accomplish the task. One generic error message covering all possible errors is not acceptable.

Your `is_same.sh` script should handle three arguments as input that indicate the two files to be compared and an optional argument. Argument one should be a filename, argument two should be a filename and argument three should be a 'T' or an 'F'. Argument three is optional, meaning if argument 3 is present and valid, your script should handle it correctly. If argument three is not present your script should still function. The third argument, if present, indicates whether to delete the log file ('T'rue or 'F'alse). Your script should check that the proper number of arguments have been passed. If there are NO command line arguments passed, then prompt the user for two files to check. If the user does not supply two valid names, print out nicely formatted warnings and exit. If there are not enough, or too many arguments, print a usage message and quit with a non-zero exits status. All messages indicating success or failure should print to the screen and be appended to a file named *log.txt*

Make sure that file names have read and write permissions set (print **all** file permission warnings out, not just the first encountered). You should check that any temporary files used during the script (if any) do not already exist. If any of these checks fail, the print an appropriate error message and exit the script with a non-zero status. If all the checks pass, then delete the file that would come last when filenames are sorted alphabetically.

Example Runs

Below are some example runs

- 'cabbage.dat' is missing permissions
- 'munkey' is a directory
- 'apple.dat' and 'Apple.dat' have identical contents

```
> is_same.sh apple.dat cabbage.dat


cabbage.dat does not have read permissions
cabbage.dat does not have write permissions
Cannot perform sameness check, please check files

> is_same.sh apple.dat munkey

munkey is not a regular file
Cannot perform sameness check, please check files


> is_same.sh apple.dat Banana.dat

Files not identical: Banana.dat not removed

> is_same.sh apple.dat Apple.dat T
Removing log.txt
File are identical:  apple.dat removed

> is_same.sh apple.dat Apple.dat Q
usage: same.sh file file [T|F]
```

Submission

Turn in your assignment, which must be named *is_same.sh*, by midnight 3/1/21 Use the *handin* program on agora.

```
handin.<course #>.<section #> <assignment #> <list of files>
handin.462.1 1 is_same.sh
```