CS 406/591 – Linux System Admin – Summer 2015

Lab #1
(Not to be handed in–just for practice)

This lab consists of a number of suggested stages of assignments to get you some practice with Bash shell scripting, as well as SED/AWK. Your practice lab(s) are not to be handed in, but solutions will be made available. We will then do a more involved shell scripting lab that will be handed in and graded.

Here is the basic application that we will experiment with: make backups of a set of files in one directory, into another directory. We want our script to be named fbackup, and it should have syntax:

fbackup SOURCE_DIR TARGET_DIR.

So fbackup is to take two command lines arguments:

- SOURCE_DIR is the directory (path) of the files to be backed up;
- TARGET_DIR is the directory where the backup copies should be put.

A valid call to the program might look like:

./fbackup ~/Documents ~/backups/Documents

You are to build several progressively more complex versions:

VERSION 1: Just to get started, you should make a very simple version that simply checks if the correct number of arguments have been given, else prints a standard usage messages and exits with failure, or if two arguments are given, it just prints out the two arguments, and exits successfully.

VERSION 2: Now modify your first version to copy everything in the source directory to the target directory. For now we will assume that all that is in the source directory are regular files. We will also assume that the target directory exists. While you could potentially just use a single cp command in your script, you are to use a for loop (mapping version), so that we will be able to operate on the files individually later. So, you want to add a for loop that will map through all the entries in the directory, and call cp on each one to make a copy in the target directory. (Hint: does you code also backup hidden files? To denote that what you have created are backup copies, you should add the extension .bak to the file copies (i.e., the copies in the target directory should have the extension .bak).

VERSION 3: Now we want to enhance your second version. It is not good practice to just assume that everything in the source directory is a regular file. For instance, you might have a FIFO, or more realistically you might have one or more (sub)directories. While you could use cp with the -r option to copy a subdirectory, you would not have the correct extensions. The first extension to make here is to add checking so that you copy only regular files. A second extension is to make your script able to deal with the target directory not existing. Add a test to see if it the target directory exists before you try to do any copying, and if it does not, create it.

VERSION 4: When doing backups, one generally wants to copy only files that need to be backed up, i.e., only those not already backed up. Extend your previous version first by only copying files from the source directory that do not exist in the target directory. You
should then try to add checking that the the target is the same as the source, and only avoid copying in that case. This is best done by using diff to compare the two files.

**VERSION 5:** For version 5, we will adjust the file names further. Let us do the following: (1) Capitalize the first character of the filename if it is an alphabetic; and (2) replace any spaces in the filename with underscores (_). To do these things, you will need to use SED or AWK. Note: do not modify the names of the files in the source directory!

**VERSION 6:** For the final version of the program, you should take version 5 and make it so that the program is able to handle (sub)directories in the source directory. That is, for each directory in the source directory, you should create a corresponding (sub)directory in the target directory, and copy any regular files in the subdirectory into the corresponding target subdirectory (with appropriate name changes). If you want to make it completely general, make your script so that it can handle arbitrary depth directories—e.g., if there is a (subsub)directory in the (sub)directory in the source directory.