Lab1 Report Documentation

Command Descriptions

- Echo – this command will send a string to the stdout (which is the terminal by default). This command can accept escape commands to allow for special characters; additionally, variables can be included but will be converted to strings.
  Example:
  echo “Hello World”
- Cp – this command will copy a file from a source to a destination. This can include subdirectories and multiple files if the -R switch is provided, which defines a recursive copy.
  Example:
  cp Source/file1.txt Destination/file1.txt
- Ls – this command will list the files in the directory specified (current directory if no directory is specified). This has different switches that can show hidden files and attributes of files as well.
  Example:
  ls -la directory
- Mv – this command will move a file from a source location to a destination location. If the location is the same for the source and destination and only the filename changes, then the file is simply renamed
  Example:
  mv source/file1.txt source/file1.bak
- Rm – this command will remove a file that is specified in the location specified. Globbing can be used to specify multiple files.
  Example:
  rm source/file1.txt
- More – this command will show the output of a file, but a screen at a time, and put the word ‘more’ at the bottom. When a key is pressed, it goes to the next screen so that you can see the information one screen at a time.
  Example:
  more source/file1.txt
- Kill – this command will delete and terminate a process listed by the process ID. You can also specify whether a process is terminated or restarted (using the -HUP command)
  Example (process ID 1234):
  kill 1234

- History – this command will show the previous commands that were entered into the shell you are running in. This is a shell program that tracks the history of previous commands and presents them as output to the terminal when run.
  Example:
  history
- () – These are the symbols that allow for array definitions. Items are included within the parentheses and separated by spaces. These are comparable to lists in other languages, and elements can be referenced by the index or numbered order in the array starting from 0.
  Example:
  Languages=(“Java” “C++”)
- $() – references output or memory space; the result is the output from the commands run inside the parenthesis.
  Example:
  echo “Hello ”$name
- $(()) – references to output that needs to be string-wrapped as there are integer operations in the parenthesis.
  Example:
  Index=$((i))
- $ – references a variable or memory space.
  Example:
  Fullname=$firstname” ”$lastname
- Date – this command will give the current date and time in a default format, but format specifiers can be listed to adjust the string output.
  Example:
  current_date=$(date +"%Y-%m-%d-%H-%M-%S")
- Time – this command will give the time it takes to run a specific command or script and report the output of that time used in real, user, and system spaces.
  Example:
  time script.sh
- Mkdir – this command will create a directory in the current directory with a name specified after the command.
  Example:
  mkdir directory1
- Rmdir – this command will delete a directory that is currently not empty. The directory must be empty for this command to work
  Example:
  rmdir directory1
- Cd – this command will change to the directory specified after the command and will use the current directory in a relative path if the full path is not specified.
  Example:
  cd file001

- For I in {}; do; done – iteration commands that allow looping through a list or a range of elements that are defined in the brackets. The commands to run in each iteration loop are between the 'do' command and the 'done' command.
  Example (3 iterations from 0 to 3):
  For I in {0..2}; do;
  echo "inside the iteration"
  done
- If []; then; else; fi – these are the commands to handle if, else conditionals. The expression in the square brackets is evaluated as true or false, and then the conditional is processed accordingly. The fi command ends the logic control for the conditional
  Example:
  if [$i -lt 10]; then echo "index is less than 10
  else echo "index is 10 or greater"
  fi
- ${[i]} – this command will reference the array inside the brackets at the index listed inside the square brackets.
  Example (the element in the languages array at the index of variable i):
  echo ${languages[$i]}
- Chown – this command will change the ownership of the file depending on the user or group that follows the command
  Example (jsummers as the user and all in the group labeled root as owners)
  chown jsummers:root lab1.sh
- Chmod – this command will change the attributes of the file depending on the attributes that follow the command
  Example (this will allow the file to be executed)
  chmod +x lab1.sh

Script Purpose

To run the script, the following command is required to execute the file:
chmod +x <script name>

This script will create a directory based on the string format for the current date and time (concatenated together). This folder will contain 10 subfolders, and in each subfolder, it will create a file that has a unique programming language listed as a string. This unique language is pulled out of an array that is the same length as the 10 subfolders that will be created.
Iteration in the for loop is used to control the commands that repeat for each subfolder and file, but different names and the index of the iteration are used to generate unique output for each subfolder, file, and language that is included in the file.