Westley Huebner

Joseph Pinrod

Tyler Clements

11/20/13

Program 4: Line Extension and Echo

For line extension and echo, we decided to have separate functions for each, as it would make them easier to work with, while simultaneously cleaning up our code and providing better organization. We made a file, named “processInput.c”, where we are storing all of our functions that are built for manipulating or analyzing the user input from the command line.

For the nshEcho() function, we are combining all of our split inputs into one temporary character array, so we can better process the individual words. We have a simple while loop checking for variable usage (denoted by the ‘@’) and adding the next word to our output variable, which is our temporary character array. After the input from the command line has been processed, it prints out the processed string.

For the lineExtension() function, we are first checking for the ‘$’ character to see if the function even needs to be used. If the character exists, a for loop finds the location of it, starting from the back, and once it finds it, holds the location of it in a integer value, “i”. The function then cuts off everything after the ‘$’ and prompts the user for the next line of input. The comment filter is then applied to the new line of input, and the old input is then combined with the new input. After all this, the function calls itself again to check to see if another ‘$’ has been entered, and if it is, proceeds to process the next line on input, and if there is not another line extension, the function returns, ending its recursive calling.