Program 2 Report

The problems that Program 2 wants us to create our own memory functions, allow for indirect aliases, and sort entries in alphabetical order. The program essentially wants us to split up each part and make it modular.

Our approach was to use linked lists for both the environmental variables and aliases. They would share a similar structure, so each would simply be an instance of the same structure. Memory would be handled as a link list as well, but it’s structure would be different from environmental variables and aliases. Each part of the insertion, delete, update, and find (all functions that were required) are designed to work with each other modularly, allowing for each to be used wherever. We decided that sorting the items in alphabetical order would be easier if they are sorted when they are first inserted. The insert function reflects that.

Our solution ended up requiring several files to keep it modular –one for environmental variable and alias structures, one for inserting, deleting, updating, etc., one for memory structure, one for memory functions, and one for the main program. An additional file was created to handle the input strings, but they are called by a function in the main file.

The most notable part of our solution is the memory. We have a memory size, a status for the block, a pointer to the next block. We added the status variable to allow for our free function to be simple. The blocks are placed one after the next in one massive array called mem.

This may be a quick summary of the program and our solution, but we hope that the comments in our code will better explain our solution. As of writing this report, the program is operational.