Casey Hale
Mohammad Sagor

Machine Problem 1

Compile

gcc main.c linked_list.c

Design

For this project, we used a combination of a user-defined node struct for ease of managing the node heads, and inserted data manually by memory location. To move the free pointer around, we type cast it to a char pointer, moved it the desired number of spaces, then recast it as a node_t pointer. To delete a node, the node before and after it are linked and it is left, unaccessible in memory.

Problems

Everything in our project works as far as we are concerned, except for two problems;

1) We never implemented acceptance from command line arguments
2) The two lines marked problem lines give us a Bus Error; it seems to have something to do with the way the stored data is being accessed.

Questions

1) Yes, memory is wasted in the delete because the data is actually just pointed around, and never removed.
2/3) This could be avoided by noting the locations of deletions, and then when a new node is inserted, insert it where the deleted node in memory was before
4) If the data being entered is bigger than the block size, that node will not be created. If anything other than 4 bit ints are passed to the functons for key and value length, the code wouldn't work. The code should work on 32 and 64 bit machines, as the command sizeof(node_t*) is used instead of stating how many bits it is.
5) The max size of the data would be 4 less bytes that in out current implementation, which uses 4 byes instead of 8 for the node pointer (because the head would take up 16 bytes of the block, instead of just 12.