Pawel Derkacz – pd304
Steffan Channer – sjc230

readme.pdf for Assignment 1

Concerning mymalloc.c

This version of malloc utilizes an implicit linked list by use of meta data in the form of an integer. The integer is read in/out by use of a union called ‘Reader’ in order to not worry about the endianess of the computer. The integer contains the size and the LSB of the integer is used to mark allocation, with 1 signify allocated and 0 unallocated. A global variable ‘unallocated’ is used to keep track of how many free bytes are left over in memory in order to help with error checks. Errors are output by use of a function called errmsg which passes the details into errprint. For free, in order to check if a block can be freed, it must be found as a source in mem and the LSB has to be 1. Once found, free changes the LSB to 0. Best fit is the algorithm used to find suitable memory chunks for allocation.

Concerning mymalloc.h

The header file contains the skeletons for mymalloc, myfree, and a function called getunalloc that returns the global variable unallocated. The file also contains the definition of errmsg for error printing, as stated before. Finally, the header file also #includes stdio and stdlib for use of printf and rand, respectively.

Concerning output

For some reason, random numbers being forced into mymalloc can cause it to break at times. I believe it has something to do with the error checks for unallocated when the memory is filled to the brim, but I am unable to pinpoint the issue. Mymalloc works most of the time, but at this point I do not have enough time to fix the issue.

Concerning testcases/testplan

Did not have enough time to implement tests E and F.