gam169

++Malloc:

Mymalloc is an implementation of malloc() and free() which checks for errors. Once detecting an error it prints the file which the call to mymalloc came from and the line number. All of the metadata and pointers returned from mymalloc() are stored within a 4096 byte array.

mymalloc():

It receives a size, filename, and line number as parameters. If the size passed is 0 it returns null and does not report it as an error. However if the size that is asked for is greater than is available it reports the error, the max size of a single pointer is 4084 since the metadata must be accounted for.

The first time mymalloc is called it creates two metadata blocks, one for the first pointer and a second to account for the remaining unused space. Anytime an unused space is large enough for the mymalloc call and an additional metadata, the block is broken up into two.

myfree():

It receives a pointer to a location in the array which was previously allocated by mymalloc. If it was not, then it is reported as an error. If the pointer was already deallocated with myfree, then it is reported as an error. If it is a null pointer, then myfree does nothing and is not reported.

Once something is deallocated myfree checks the neighboring blocks, if they are also free then they are merged.

Metadata:

Metadata is held in a struct that contains a short int which accounts if it is being used or not, a short int which holds the size, and a pointer to the next metadata.

Using pramga pack(1) this allows the for padding to be removed as it can be placed at any boundary. This allows for the metadata size to be equal to 12 bytes, since each short int is 2 bytes and the pointer is 8 bytes. While this might result in alignment issues, it was done for the purpose of lowering the metadata size.

printMem():

Prints the values for each byte of the entire 4096 byte block.

memgrind.c:

Contains the 5 workloads located within the main method which are done 50 times. Each iteration is timed and recorded. Each time is printed for the corresponding workload iteration as well as the overall time and the average time. The average time was below a second when testing using the iLab machines.

mymalloc.h:

Header file which contains the struct, the functions, and the Defines which replace malloc and free calls to mymalloc and myfree calls.

Additional Defines:

DEBUG = 1 will print out information that can be useful for debugging.

CLEAN = 1 will replace the indexes of a pointer to 0 whenever it is freed. This makes it much easier to tell what is metadata and what is the pointer when printing memory values.