Asst1

@Thurgood Kilper

@Andrew Pagano

Mymalloc.c- my malloc creates a linked list which keeps track of how many malloc and free calls there have been. The head of the list is equal to the beginning of the 5000 sized array myBlock. Malloc() initializes the head, iterates through the list until it finds a free block with the requested size or a free block with more than the requested size. If it finds a larger block, it calls splitBlock() which creates a new block to hold the remaining free memory. Free() checks if the pointer requested to be freed is within bounds, and not already free. Then it frees the block, and calls combineFreeBlocks() which iterates through the list and combines and sequential free blocks.

Memgrind.c- Tests A-F are found here. We run each test by calling freePlus (free) and mallocPlus(malloc) which is found in mymalloc.c and utilized through mymalloc.h. In the main{} we run each test 100 times and output the average time with gettimeofday.

We tested for all possible errors.

We figured C and D take longer than A and B because the memory being allocated is larger, and there are more instances of memory not fitting in myBlock.

Average Cases: (GetTimeOfDay)

A: 0.036112

B: 0.022409

C: 0.038369

D: 0.043093

E: 0.003466

F: 0.006896