

Daniel Schley
Ryan Dunn

This program simulates malloc and free by providing memory placed on a "heap" which is a static 5000 character array. Malloc and free also provide error messages when memory cannot be allocated, when trying to allocate 0 or less bytes, trying to free an offset pointer, trying to free a pointer that was never allocated, and trying to free a pointer that was already freed.

The memory was kept track of on the heap with structs called blocks that keep track of the previous block, next block, size of the space allocated for the pointer, whether the pointer was freed, and an id to make ensure that you are looking at a block instead of some offset part of memory. The id, size, and open fields are all short integers in order to avoid using unnecessary extra space.

When using malloc, new blocks of memory replace freed memory blocks if they can fit between other memory blocks, gets inserted in space between two memory blocks if there is enough room, else it is put at the end of the current list of memory blocks where there is room.

When using free, the block that holds the details of the pointer to be freed has its open bit flipped to represent open. If there are adjacent free blocks, they will merge together in order to open up space for more allocation.