The data structure I used to track each memory allocation was a link list. There are three general cases to keep track when allocating memory. The first is when the link list “heap” has not been initialized, so the link list is empty. this causes the first call to sbkr and initializes the “heap”, link list. The second case is when there is enough heap space for the new allocation, in this case the first fit algorithm was used to find where there was sufficient space. Looping from the head of the link list, the algorithm would check each nodes size and its usage(whether its being use 1 or its free memory 0). Once a node with sufficient size is found, it get split into two nodes, the first node is the node to be returned and can be use, the second node is the remaining memory. The allocated nodes are at the front since they can potentially be reused when the node of a common size is freed, and by being at the front the heap(link list) has faster access to them. The third and last case is when the head does not have sufficient space for the required allocation size, in this cases the link list will be completely traversed, once it get to the end will cause a call to sbrk extending the “heap”.