Malloc

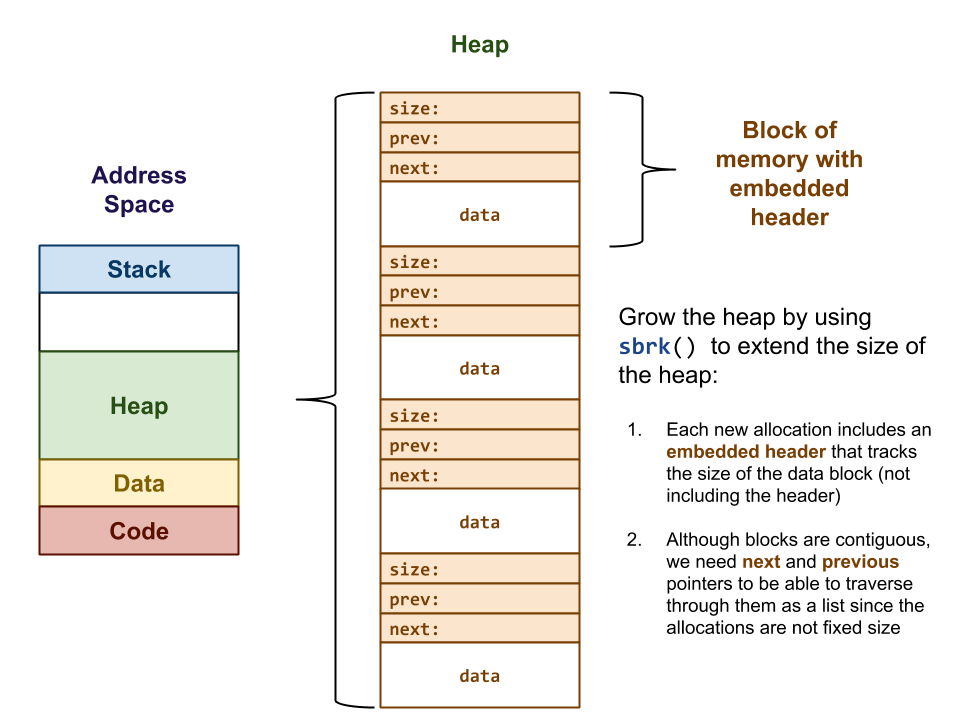
Mark Nakamae

Cal Poly SLO

April 2020

Description

This project involved writing my own malloc, calloc, realloc, and free methods using sbrk to manually request for blocks of memory from the operating system. My program would then take this block of memory, and apply a linked list structure on top of it, in which chunks of memory were given to the user as they requested more dynamic memory. The chunks were in approximately blocks of 4kB, and were aligned for both 32-bit and 64-bit systems. Here is a very helpful image explaining how the heap is formatted given the linked list structure:



Like the original implementation of free and malloc, my free would allow a chunk of memory to be reused for dynamic memory allocation without necessarily clearing the data at this chunk. This project reinforced heap structure, and what happens behind the scenes for dynamic memory allocation.

The test directory has a small program that requires dynamic memory allocation, and uses my malloc instead of the default library’s malloc.