Eklavya Patel
Krishna Chamarti
Systems Programming
SPRING 2019
Malloc++

The purpose of this project is to mimic the behavior of malloc() and free() on a virtual heap of 4096 bytes. In order to do this, we use a structure to store metadata properties that help locate the data on the heap.

METADATA STRUCTURE DESCRIPTION:

void* ptr — pointer to the data of the block, returned to user
int size — integer value of the memory in bytes
char inUse — value denoting wether or not the block is being used
'y' for yes. 'n' for no
struct metadata* next — link to the next block in the array

FUNCTION DESCRIPTIONS:

currIndex: Iterates through the array and counts the space before the given allocation, returning the index at which the metadata starts.

firstMalloc: Returns bool indicating if malloc has not been called yet.

mymalloc: allocates space for the gives number of bytes in the static array using a first fit algorithm.

merge: finds two consecutive blocks that are both not currently being used and merges them. Then checks to see if there is a third consecutive block that is not in use and if so, merges it.

myfree: Frees the memory allocation of the given pointer, if possible.

TESTING AND INSTRUMENTATION:

Average Run Times for workloads tested 100 times each

TEST A: 3 microseconds
TEST B: 149 microseconds
TEST C: 185 microseconds
TEST D:

TEST E:
TEST F: