Aaron Bruner

Dr. Jon Calhoun

ECE-2230

2 April 2021

**Test Plan**

The primary goal for our test plan is to demonstrate that our functions are working properly as expected. Thus, these test cases below will demonstrate that our newly created memory management functions work as expected.

We tested our code in this MP by creating unit test drivers in lab4.c. We created a total of 7 drivers all demonstrating separate things.

* -u 0: This test is a provided test which allocates 16 bytes of memory and then frees it.
* -u 1: This test is a provided test that makes 4 allocations including a new page.
* -u 2: This test requests for four chunks of memory and then demonstrates the BEST\_FIT option
* -u 3: This test will allocate multiple blocks of memory that require several pages to fulfill
* -u 4: This test creates a very disorganized free list by allocating and freeing over and over. We can watch how coalescing will keep everything nice and clean.
* -u 5: This test attempts to demonstrate how when coalescing the blocks are combined and contain an entire page of memory. The idea is that when we call Mem\_alloc we will not need to get a whole new page; instead, use memory in the free list.
* -u 6: This test is designed to show that coalescing will combine the free memory when it can. So, for this demonstration we create a page of size 1/5, 1/6, 1/15, and 1/16 and when we remove both 1/6 and 1/15 we should see those two blocks of memory be combined in the free list.

It is clear that our code is at least managing the memory correctly since valgrind and our own memory stats report that all memory is in the heap and no leaks are possible.