Nicholas Jordan, Bryce Holley, Lei Wang

June 2, 2014

# CS 411 Project 4

## Our Solution

To implement the SLOB best-fit algorithm, we first analyzed how the existing first-fit implementation works. What first-fit does is search through a list of pages, finds the first page with enough free space, and then searches within that page for the first available block to allocate. Our best-fit implementation keeps the search through the list of pages, finds the first page with enough free space, but within that page it searches for the block with closest fit, at which point it allocates. To enable SLOB, we used make menuconfig, general setup, and selected SLOB.

Since the slob_page_alloc() function controls the search within a page, this is the function we modified. Instead of immediately allocating at the first available block, we moved all relevant allocation code to run only at the end of the loop, after a best is found. When an available block is found, we have a secondary check to see if the difference is smaller than the previous best difference. This will result in having the best difference at completion of the loop. The saved best variables are prev, cur, aligned, delta, and the difference.

## Testing

We tested the fragmentation of both implementations by introducing global variables to keep track of the total claimed size of the slob (page size * number of pages created) as well as the amount used of that claimed space, such that the difference (claimed - used) is the amount of free/unused space in the allocated slob. In order to test the slob performance, we added two system calls: sys_get_slob_amt_claimed , sys_get_slob_amt_free. This allows us to call it in user space, and return the claimed and used variables. We then print the percentage of space that is unused (free/claimed) to present a more portable metric with memtest.c.