Mykola Gryshko & Joshua Kim
Operating Systems: Asst2 Write Up

Rather than using a traditional 8MB static character array, our implementation used memalign() to request a 8MB block and 16MB swap space. The 8MB block is partitioned in a way such that approximately 2MB is dedicated to OS space, and 6MB is dedicated to the users [threads]. The swap space is used when a page table needs to swap out existing pages with another thread's pages.

When malloc is passed with a 'LIBRARYREQ' argument, that indicates that malloc is used to allocate memory for data structures used by the thread library such as the scheduler, queues, etc. When malloc is passed with the 'THREADREQ' argument, that indicates that malloc is used to allocate memory requested through the user threads.

Memory allocation is handled in sync with the page table and memory book. When memory is requested, the memory book is traversed to check to see if valid, and if so, the head of the memory entry struct is retrieved. If it does not exist, a new head is created and returned, or no more pages are available, in which pages would have to be swapped via the swap space.

Once the head is retrieved, it is traversed via the findBestFit() function to find a block of memory that is the best fit for the requested amount of memory. If there is no more memory left on the page, another page must be requested for more memory [since a page has approximately 4KB of memory]

The free function sets every allocated memory entry (that is no longer used) to be available when a thread requests for memory. At the end of a free call is a coalesce, which traverses the head and combines any contiguous free blocks into one larger one.