CS5250 - Assignment 4

Daniel Alfred Widjaja - A0184588J

Section A

- I implemented the buddy allocator with 2-LRU list.
- There are 3 operation that could be implemented
    - Allocate
        - I put all the pages in the LRU inactive list
        - Remove all evicted pages from buddy system
        - Try to put everything continuously in the buddy system
        - If there's no space, I will allocate the page by page in the buddy system
    - Access
        - 3 possible cases:
            - The page is in inactive list
                - Remove from inactive list
                - Insert to active list
            - The page is in active list
                - Nothing to do
            - The page is not in both
                - Will insert to LRU and evict a page if needed.
                - If any page is evicted, then remove from buddy allocator
                - Insert to buddy system
    - Deallocate
        - If the page is in inactive list
            - Remove from inactive list
            - Remove from buddy allocator
        - If the page is in active list
            - Remove from active list
            - Remove from buddy allocator
        - If the page is not in both
            - Nothing to do
- The output format will consist of 3 parts,
    - Buddy allocator
        - The output will have
            - Page ID
                - The id of the occupied page
            - Group ID
                - Page with the same group ID are merged together
            - Length of group
                - The number of page with the same group ID
            - Seqno

        - num
- Inactive LRU
    - The output will have
        - Index
            - Index of the element in the list
            - if the list is full, will evict the one with id 0
        - seqno
        - num
- Active LRU
    - The output will have
        - Index
            - Index of the element in the list
            - if list is full, will move element with id 0 to inactive list.
        - seqno
        - num

Section B

1.
    a. Page0 = 0 | 0 | 4 = 4
    b. Page1 = 2 | 63 | 1 = 2 * 64 * 64 + 63 * 64 + 1 = 12225
    c. Page2 = 63 | 62 | 1 = 63 * 64 * 64 + 62 * 64 + 1 = 262017
2.
    a. init int *shared_ptr = 0;
    b. spin_lock() {
        i. while (!cas(shared_ptr, 0, 1)) {

            }
    }
    c. spin_unlock() {
        i. cas(ptr, 1, 0);
            }
    }