# End semester OS lab exam Description

AU1940127

Jay Ghetia

## Question 1

## Question 2

**Allocation in Binary Buddy System**

- The buddy system maintains a list of the free blocks of each size (called a free list), so that it is easy to find a block of the desired size, if one is available.
- If no block of the requested size is available, allocate searches for the first nonempty list for blocks of at least the size requested.
- In either case, a block is removed from the free list.
- This process of finding a large enough free block will indeed, be the most difficult operation for us to perform quickly.
- If the found block is larger than the requested size, say 2^k instead of the desired 2^i, then the block is split in half, making two blocks of size 2^k-1.
- If this is still too large (k - 1>i), then one of the blocks of size 2k-1 is split in half.
- This process is repeated until we have blocks of size 2^k-1, 2^k-2, ..., 2^i+1, 2^i, and 2^i.
- Then one of the blocks of size 21 is marked as occupied and returned to the user.
- The others are added to the appropriate free lists.
- Each block B1 was created by splitting another block into two halves, call them B1 (Buddy of B2) and B2(Buddy of B1).

**Deallocation of Binary Buddy System**

- Now when a block is deallocated, the buddy system checks whether the block can be merged with any others or more precisely whether we can undo any splits that were performed to make this block.
- The merging process checks whether the buddy of a deallocated block is also free, in which case the two blocks are merged;
- then it checks whether the buddy of the resulting block is also free, in which case they are merged; and so on.

## Question- 3

The Producer-Consumer problem is a well-known challenge for multi-process synchronisation, or the synchronisation of several processes.

In the producer-consumer issue, there is one Producer who produces things, and there is one Consumer who consumes the Producer's stuff. The producers and consumers share the same fixed-size memory buffer.

The Producer's role is to produce data, store it in the buffer, and then generate data again. The Consumer's task is to consume the data from the buffer.

What exactly is the issue here?

- When the buffer isn't full, the producer should just generate data. If the buffer is full, the producer should not be able to add any more data to it.
- When the buffer is not empty, the consumer should consume data. The consumer should not be able to take any data from the buffer if it is empty.
- The buffer should not be used by both the producer and the consumer at the same time.