CSE-332 –Operating System

End semester exam

Name: Umang Patel **Enrollment No.**: AU1940174

Section: 2

---

1) We have to done this question using multithreading. Here we used 3 threading to print a desire pattern.

2)

b)

In the buddy's algorithm, the size of an empty block is given in the form of an integer power of 2. As a example 2,4,8,16,32 up to the memory size. When a free block of size 4k is requested, a free block is allocated from the list of free blocks of size 4k. If an empty block of size 4k is not available, the next largest block 4k + 1 is split in half this method is known as a buddy's algorithm which is able to meet the demand of the requests.

Example:

Let take one memory size of 1024KB and we take a processor P which requires 90KB swap. 128KB is sufficient, as the hole list is dedicated to powers of 2. Initially (when process start) there are no any 128KB available so firstly 1024KB block is split into two buddies of 512KB each, it is further spilt into two buddies of 256KB each, still it is further split into two buddies of 128KB blocks and now we reach the condition so one of that is allocated to the process. Now if we take processor P2 which requires a 49KB thus we required 64KB of block. Therefore, when 128KB block split into two 64KB buddies then if we take one more process P3 which required 135KB then that will be adjusted in whole 256KB block. So, at the end if any such blocks are free then they block recombined and can make one larger original block.

Output:

```
umang@umang:~$ g++ buddy_allocation.cpp
umang@umang:~$ ./a.out
Memory from 0 to 31 allocated
Memory from 32 to 39 allocated
Memory from 64 to 127 allocated
Failed to allocate memory !!!
umang@umang:~$
```

Buddy Allocation

```
umang@umang:~$ g++ buddy_deallocation.cpp
umang@umang:~$ ./a.out
Memory from 0 to 15 allocated
Memory from 16 to 31 allocated
Memory from 32 to 47 allocated
Memory from 48 to 63 allocated
Memory block from 0 to 15 freed
invalid free request !!
Memory block from 32 to 47 freed
Memory block from 16 to 31 freed
Blocks starting at 0 and 16 was done
```

Buddy Deallocation

Q-3)

The producer consumer problem is a synchronization problem. There is a fixed size buffer, and the producer creates an item and puts it in the buffer. Consumers delete item from the buffer and consume them. If the consumer is consuming an item from the buffer, the producer should not generate the item in the buffer and vice versa. As the producer or consumer needs access to the buffer.

Producer consumer problem can be solve using Semaphores and Mutex.