

Name: Nirav Karavadra

Roll no.: AU1940198

CSE332 – Operating System (Section-2)

End Semester Exam

Que 1.

Print the following Pattern “A 1 a B 2 b C 3 c ... Y 25 y Z 26 z” Using any one of the following concepts a. Multiprocesses b. Multithreads

The problem is to synchronize n number of threads using pthread library. The idea is take thread count and print first thread, print second thread and print third thread.

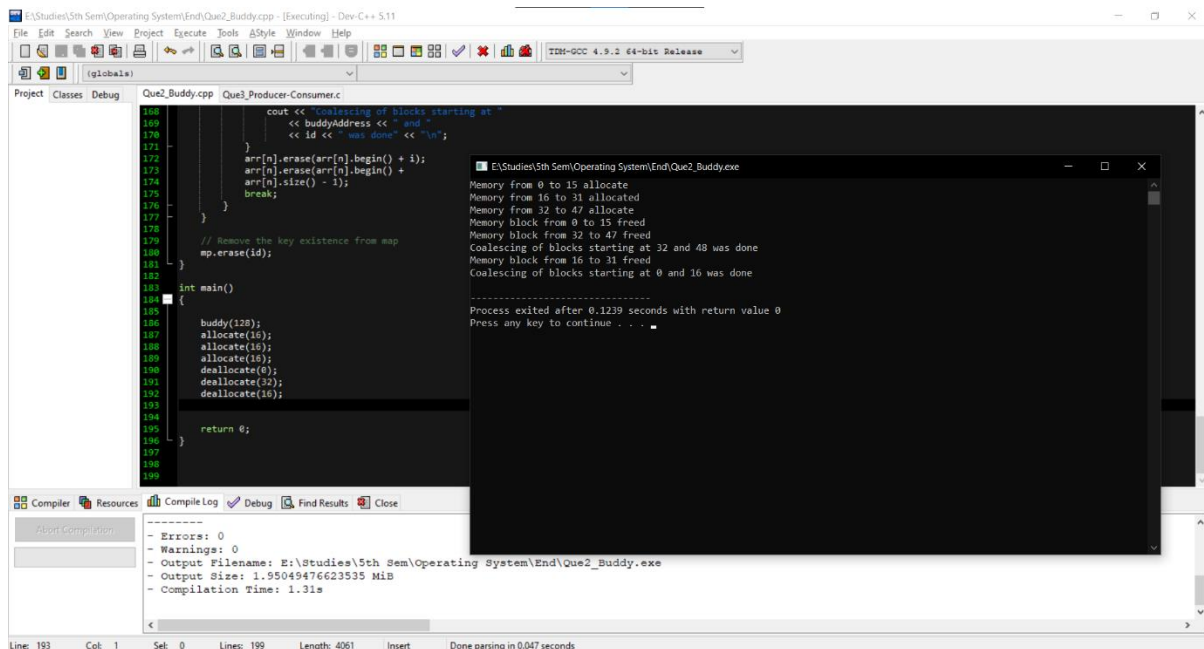
Que 2.

Describe and implement any one of the following

b. Describe the Buddy’s Algorithm for Memory Allocation and Deallocation along with an example and implement it in C or C++.

In buddy system there are sizes of free blocks in form of integral power of 2. For example, 2, 4, 8, 16 etc. When a free block of size 2^k is requested, from the list of free blocks of size 2^k is allocated. If no free block is available, the block of next larger size that is 2^{k+1} is split in two halves called buddies to fulfil the request.

There is total memory size of 512KB and let a process P1, requires 70KB to be swapped in. As the hole lists are only for powers of 2, 128KB will be big enough. Thus, 512KB block is split into two buddies of 256KB each, and one further split into two 128KB blocks and one of them is allocated to the process. Next P2 requires 35KB. Rounding 35KB up to a power of 2, a 64KB block is required. So, when 128KB block is split into two 64KB buddies. After satisfying the request in this way when such block is free, the two blocks/buddies can be recombined to form the twice larger original block when it is second half buddy is also free.



Que 3.

Describe what is Producer Consumer Problem and its solution in detail using Semaphores and Mutex and implement it in C.

The Producer-Consumer problem is used for multi-process synchronization, i.e., synchronization between multiple processes. In the producer-consumer problem, one Producer produces something, and one Consumer consumes that product. The same memory is being shared by both. The Producer's job is to generate the data, put it into the buffer, and start generating data again. At the same time, the position of the Consumer is to consume the data from the buffer.

In the producer-consumer problem, we use three semaphore variables. *Semaphore S*: This semaphore variable is used to achieve mutual exclusion between processes. By using this variable, either Producer or Consumer will be allowed to use or access the shared buffer at a particular time. This variable is set to 1 initially. *Semaphore E*: This semaphore variable is used to define the space in the buffer. Initially, it is set to the whole space of the buffer, i.e., "n" because the buffer is initially empty. *Semaphore F*: This semaphore variable is used to define the space that is filled by the Producer. Initially, it is set to "0" because there is no space filled by the Producer initially.

