Question 3: -

3.1: What is a semaphore? How is it different from a mutex?

Ans: A Semaphore is a variable used to control access to common, shared resources that need multiple processes by performing wait () and signal () operations. There are two types of semaphores;

1) Binary semaphore - state variable can only be 0 or 1.

2) Counting semaphore - state variable can have non-negative integer values.

Mutex (mutually exclusive) is an object owned by a thread, It allows only one thread to access resources. Whereas in semaphore is a signaling mechanism, so it allows a number of threads to access shared resources.

3.2: What is the difference between a task and a thread in an RTOS?

Ans: A task is a semi-independent portion of the application that carries out a specific duty, and a task tends to be a high-level duty that is carried out by the application.

Thread is a semi-independent program segment that executes within a process. A thread is then a small program that has a specific function and purpose within the overall application.

Threads tend to be lower level than tasks.