MUTEX VS SEMAPHORE

Use of Semaphore

In the case of a single buffer, we can separate the 4 KB buffer into four 1 KB buffers. Semaphore can be associated with these four buffers. This allows users and producers to work on different buffers at the same time.

Use of Mutex

A mutex provides mutual exclusion, which can be either producer or consumer that can have the key (mutex) and proceed with their work. As long as producer fills buffer, the user needs to wait, and vice versa. In Mutex lock, all the time, only a single thread can work with the entire buffer.

KEY DIFFERENCE

1. Mutex is a locking mechanism whereas Semaphore is a signaling mechanism
2. Mutex is just an object while Semaphore is an integer
3. Mutex has no subtype whereas Semaphore has two types, which are counting semaphore and binary semaphore.
4. Semaphore supports wait and signal operations modification, whereas Mutex is only modified by the process that may request or release a resource.
5. Semaphore value is modified using wait () and signal () operations, on the other hand, Mutex operations are locked or unlocked.

Parameters	Semaphore	Mutex
Mechanism	It is a type of signaling mechanism.	It is a locking mechanism.
Data Type	Semaphore is an integer variable.	Mutex is just an object.
Modification	The wait and signal operations can modify a semaphore.	It is modified only by the process that may request or release a resource.
Resource management	If no resource is free, then the process requires a resource that should execute wait operation. It should wait until the count of the semaphore is greater than 0.	If it is locked, the process has to wait. The process should be kept in a queue. This needs to be accessed only when the mutex is unlocked.
Thread	You can have multiple program threads.	You can have multiple program threads in mutex but not simultaneously.

Ownership	Value can be changed by any process releasing or obtaining the resource.	Object lock is released only by the process, which has obtained the lock on it.
Types	Types of Semaphore are counting semaphore and binary semaphore.	Mutex has no subtypes.
Operation	Semaphore value is modified using wait () and signal () operation.	Mutex object is locked or unlocked.
Resources Occupancy	It is occupied if all resources are being used and the process requesting for resource performs wait () operation and blocks itself until semaphore count becomes >1.	In case if the object is already locked, the process requesting resources waits and is queued by the system before lock is released.