

Implementation of Synchronization using a Counting Semaphores in microcontroller

Dr.Ir. Widi Setiawan

- Just as binary semaphores can be thought of as queues that have a length of one, counting semaphores can be thought of as queues that have a length of more than one.
- Each time a counting semaphore is 'given', another space in its queue is used. The number of items in the queue is the semaphore's 'count' value.

Counting semaphores are typically used for two things:

1. Counting events

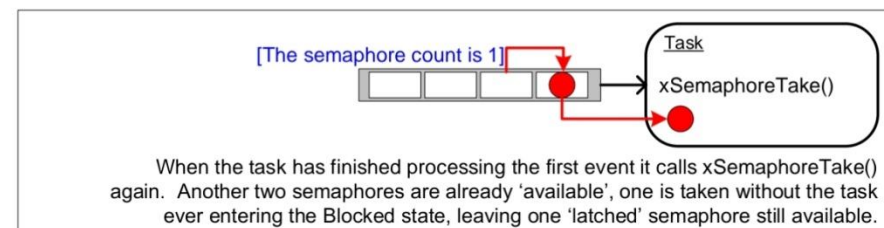
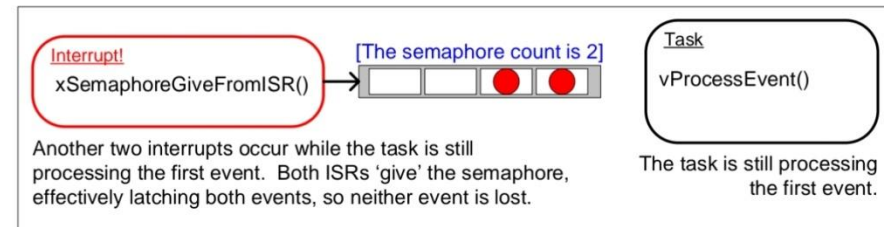
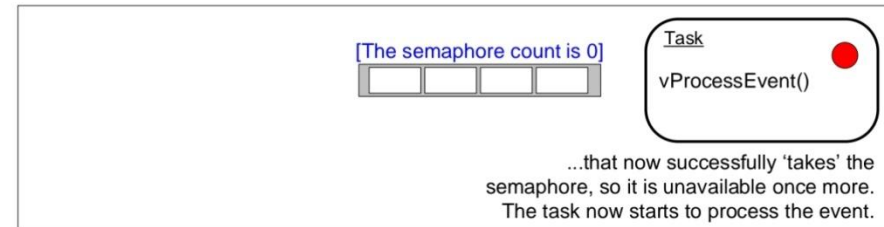
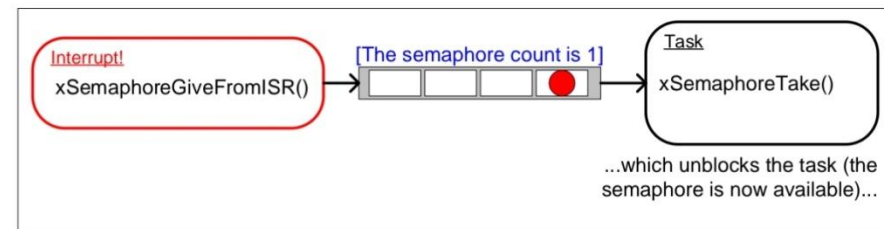
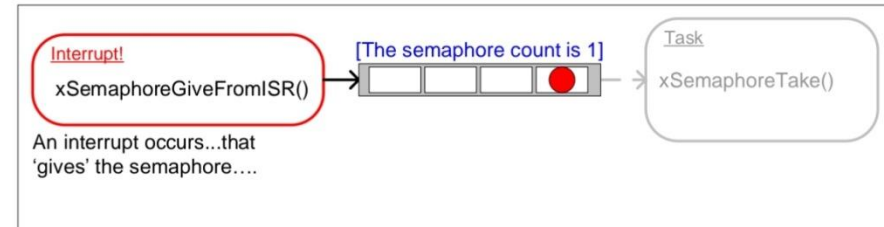
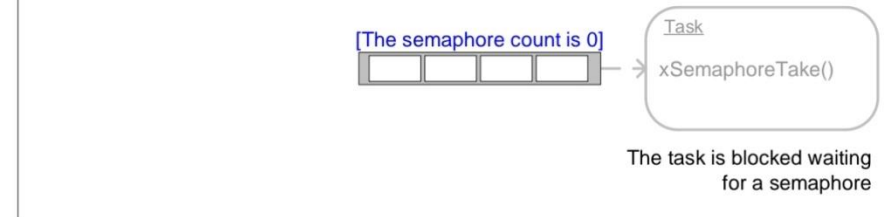
- In this scenario, an event handler will 'give' a semaphore each time an event occurs— causing the semaphore's count value to be incremented on each 'give'.
- A task will 'take' a semaphore each time it processes an event—causing the semaphore's count value to be decremented on each 'take'. The count value is the difference between the number of events that have occurred and the number that have been processed. This mechanism is shown in Figure in the next slide.
- Counting semaphores that are used to count events are created with an initial count value of zero.

2. Resource management.

- In this scenario, the count value indicates the number of resources available.
- To obtain control of a resource, a task must first obtain a semaphore— decrementing the semaphore's count value.
- When the count value reaches zero, there are no free resources.
- When a task finishes with the resource, it 'gives' the semaphore back— incrementing the semaphore's count value.

Synchronization using a counting semaphore

Using a counting semaphore to 'count' events



- SemaphoreHandle_t
xSemaphoreCreateCounting(UBaseType_t
uxMaxCount,
UBaseType_t uxInitialCount);

uxMaxCount

- The maximum value to which the semaphore will count. To continue the queue analogy, the uxMaxCount value is effectively the length of the queue.
- When the semaphore is to be used to count or latch events, uxMaxCount is the maximum number of events that can be latched.
- When the semaphore is to be used to manage access to a collection of resources, uxMaxCount should be set to the total number of resources that are available.

uxInitialCount

- The initial count value of the semaphore after it has been created.
- When the semaphore is to be used to count or latch events, uxInitialCount should be set to zero—as, presumably, when the semaphore is created, no events have yet occurred.
- When the semaphore is to be used to manage access to a collection of resources, uxInitialCount should be set to equal uxMaxCount—as, presumably, when the semaphore is created, all the resources are available.

Returned value

- If NULL is returned, the semaphore cannot be created because there is insufficient heap memory available for FreeRTOS to allocate the semaphore data structures.
- A non-NULL value being returned indicates that the semaphore has been created successfully. The returned value should be stored as the handle to the created semaphore.