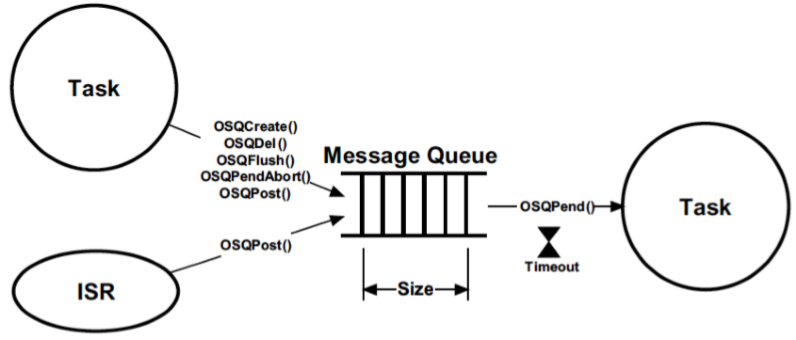
Message Passing

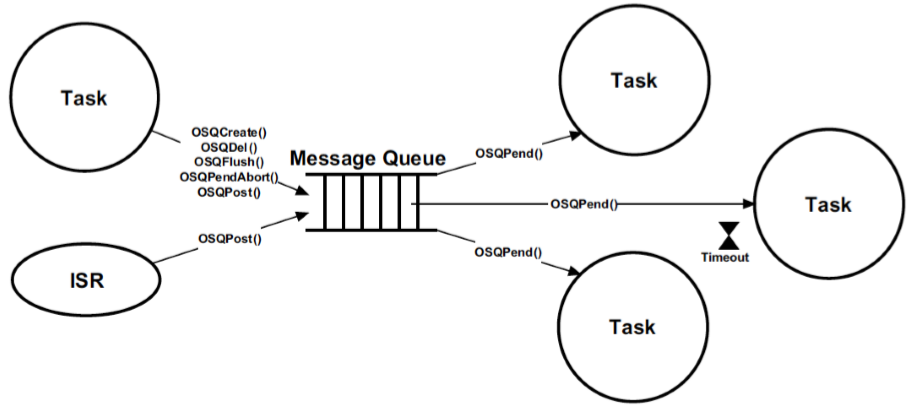
(\*)Message queues are drawn as a first-in, first-out pipe (FIFO). However, with C/OS-III, it is possible to post messages in last-in, first-out order (LIFO). The LIFO mechanism is useful when a task or an ISR must send an “urgent” message to a task

1. uCOS 提供message Q 有last in first out的功能.



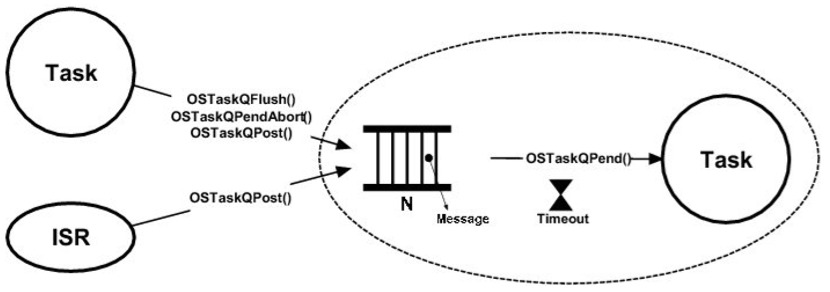
(\*)The message queue also contains a list of tasks waiting for messages to be sent to the message queue

1. Message Q也有像semaphore一樣,有broadcast的功能



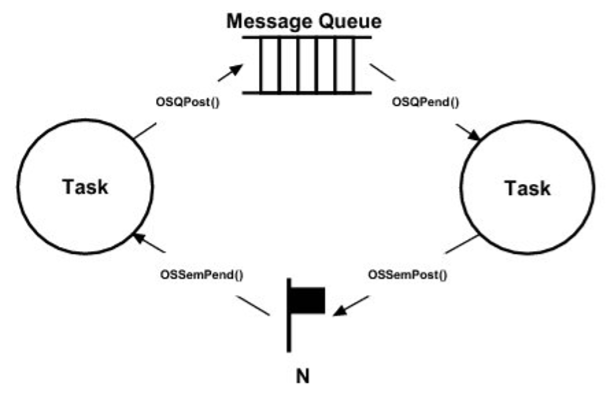
(\*)A message queue is built into each task and the user can send messages directly to a task without going through an external message queue object

1.每一個task有自己built in的message Q; 如同buil-in的semaphore一樣; 可以省去宣告external的message Q



(\*)One task produces data while the other it. However, data processing takes time and consumers consumes might not consume data as fast as it is produced. In other words, it is possible for the producer to overflow the message queue if a higher-priority task preempts the consumer.

1. 產生速度, 低於消耗速度, 用一個counting semaphore來防止 Q overflow.



Producer Task:

Pend on semaphore;

Send message to message queue;

Consumer Task:

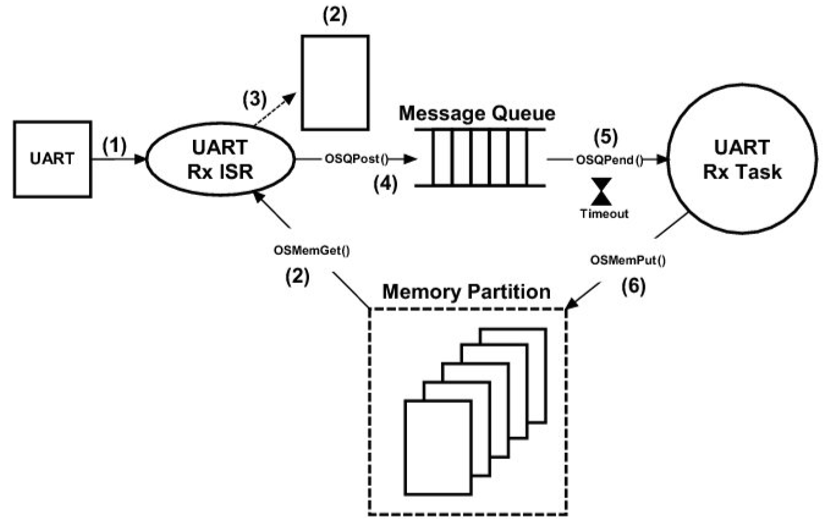
Wait for message from message queue;

Signal the semaphore;

搭配 Resource Management和 Message Q的操作:

(\*)The messages sent typically point to data structures, variables, arrays, tables, etc. However, it is important to realize that the data must remain static until the receiver of the data completes its processing of the data.

1. Data在傳遞的過程當中, 必須維持靜態的(Static)直到reciever處理完畢



(1) UART接收charater產生interrupt

(2) ISR會先從memory partition取出一塊buffer, 此buffer會在interrupt發生時填入資訊,直到填滿為止

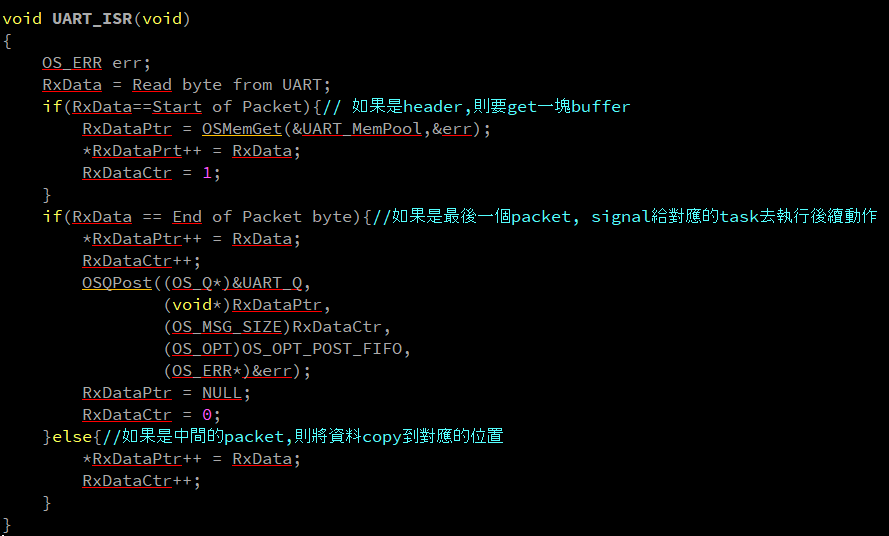
(3) 持續將接收到的資料填入buffer中,直到填滿為止

(4) buffer填滿之後(EOP),會發出一個QPost的 signal,通知需要此information的task

(5) Rx task在得到此post之後,狀態會遷移到ready, scheduler重新排程若為highest prioirty, 則run

(6) Rx task在執行完之後, 將之前的buffer release回memory partition.

範例程式:



(\*)Another interesting use of message queues is shown in he figure below. Here, a task (the t server) is used to monitor error conditions that are sent to it by other tasks or ISRs (clients).

(\*)The message sent indicates the error detected, which threshold was exceeded, the error code that is associated with error conditions, or even suggests the address of a function that will handle the error, and more

1. Client和server的關係,可用message Q來完成; Error Handler task為一server, 接收來自其他client如 task,ISR所傳出的資訊

2. Message內容是 一個variable, 也可以是一個pointer to function來執行 error的後續動作

