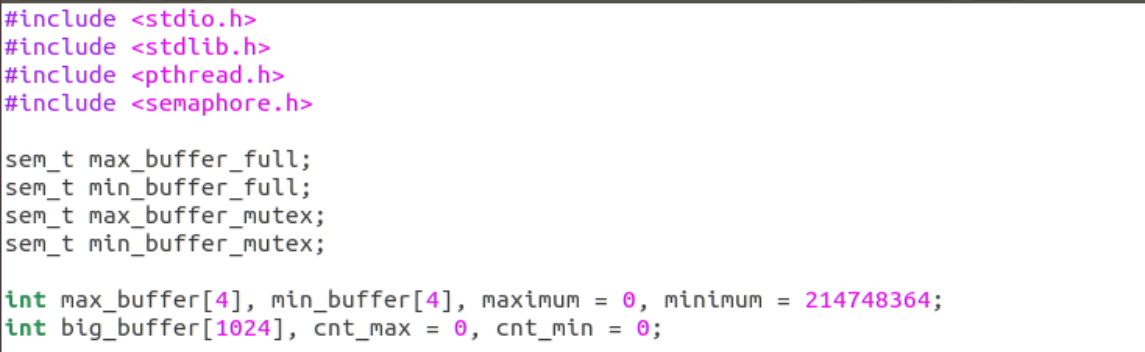
**程式碼**

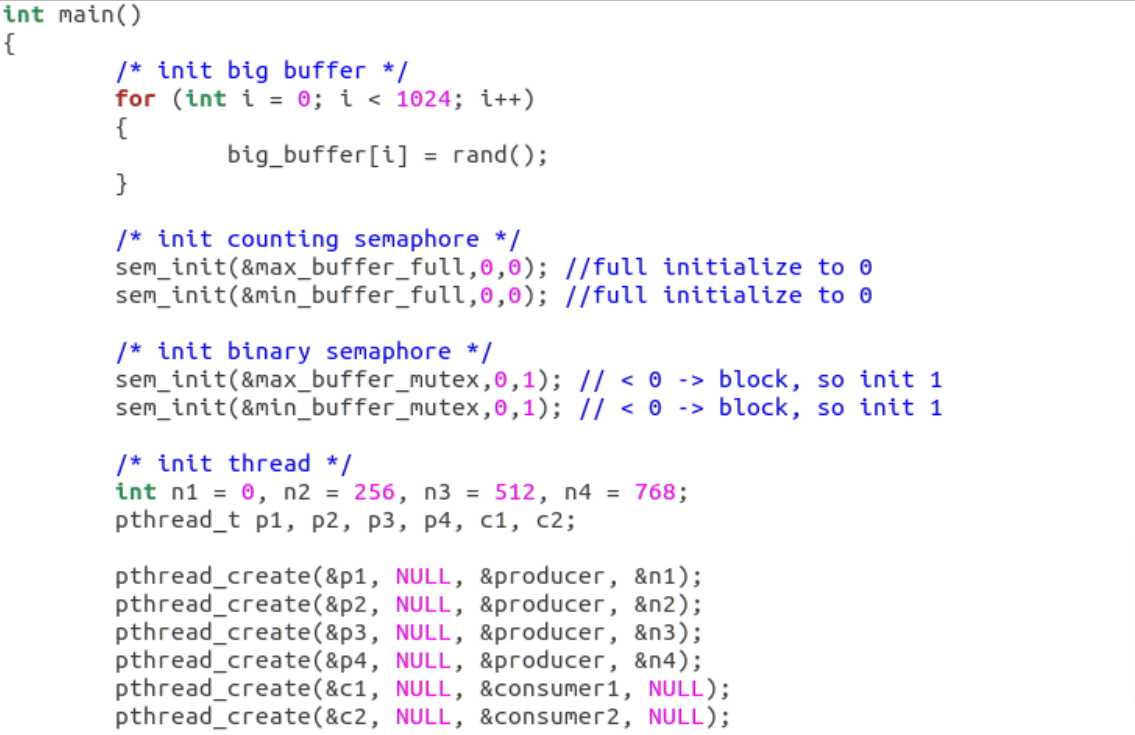
1. **全域變數**

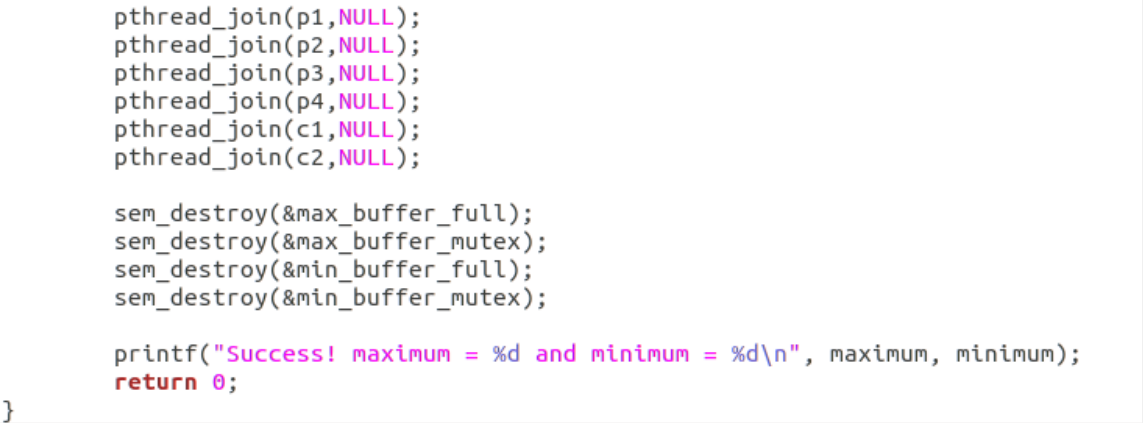
****

由於有max\_buffer和min\_buffer兩個shared data(p.s以及他們的index，cnt\_max與cnt\_min)，所以我們會建立兩個binary semaphore來做存取控制。

另外，因consumer會去取用buffer的資料，所以會建立兩個semaphore來確保consumer不會去存取到空的資料。

1. **main function**



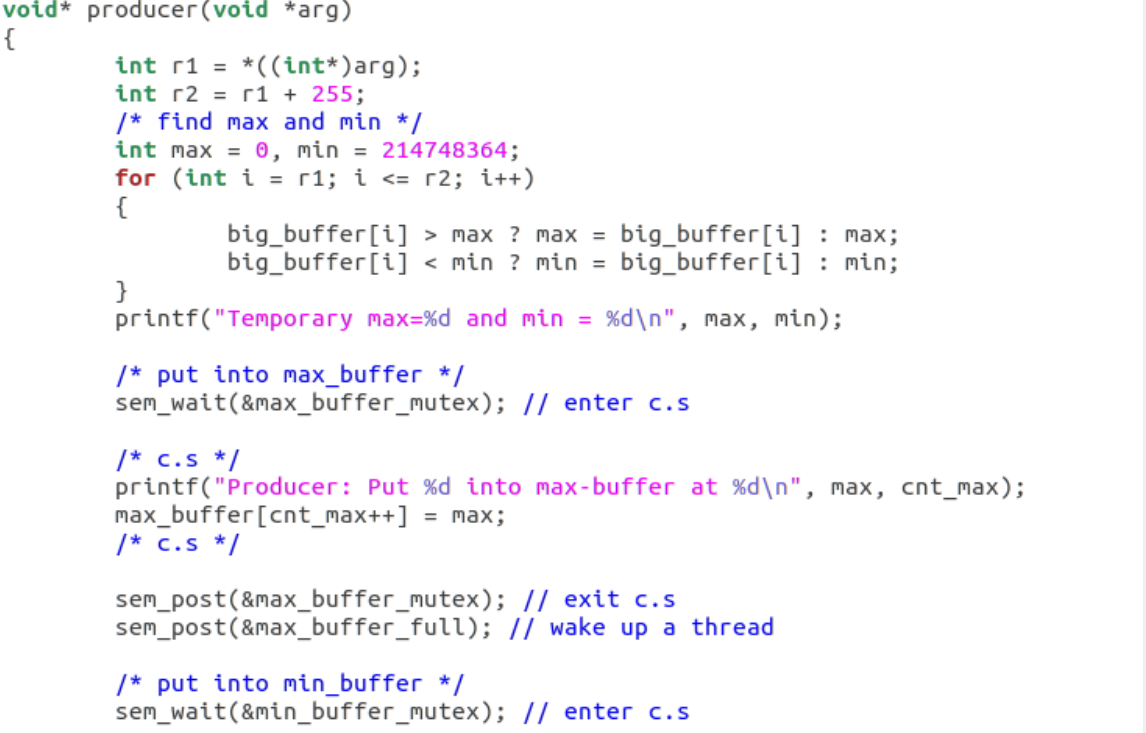


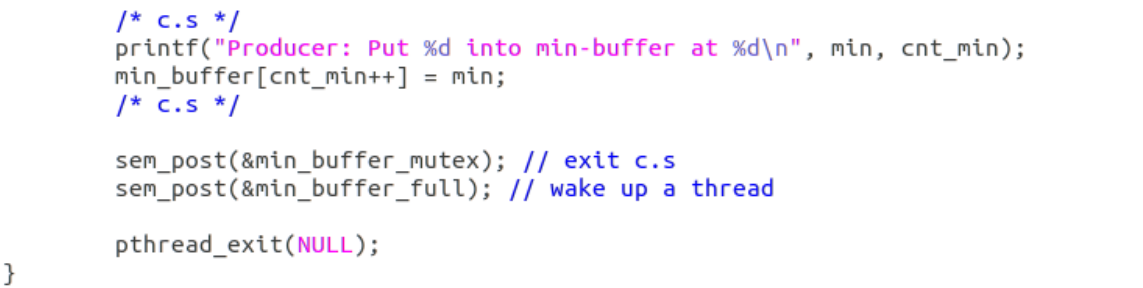
在main中會先初始化各項資料(big\_buffer中的亂數值，semphore的建立設定等等)

之後建立4個thread，p開頭是producer，會依照參數執行producer(function說明在下面)；而c開頭的是consumer，consumer1用來找maximum，consumer2則用來找minimum。

最後執行完畢會回收semphore並印出最後的答案。

1. **producer function**

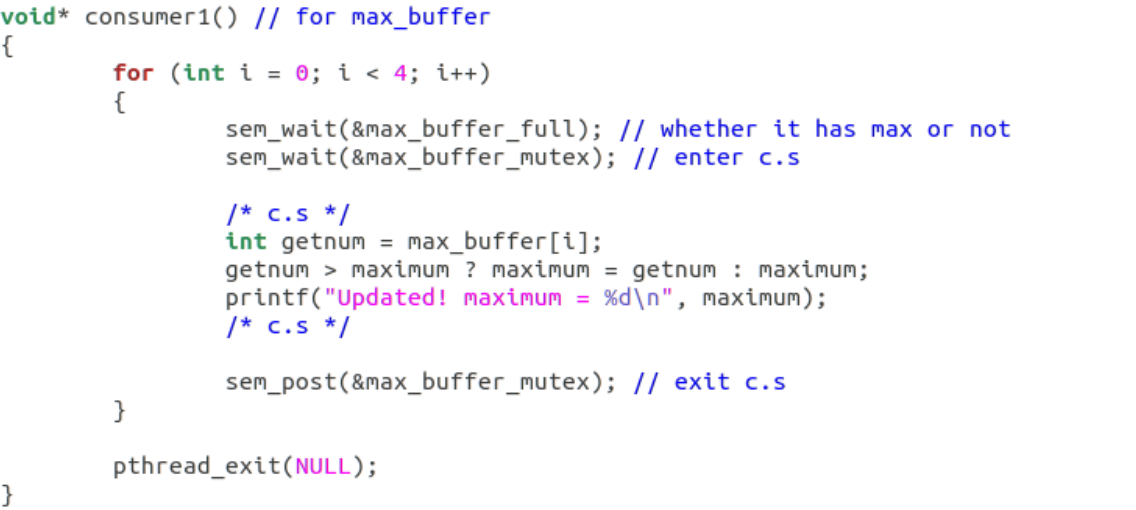
****

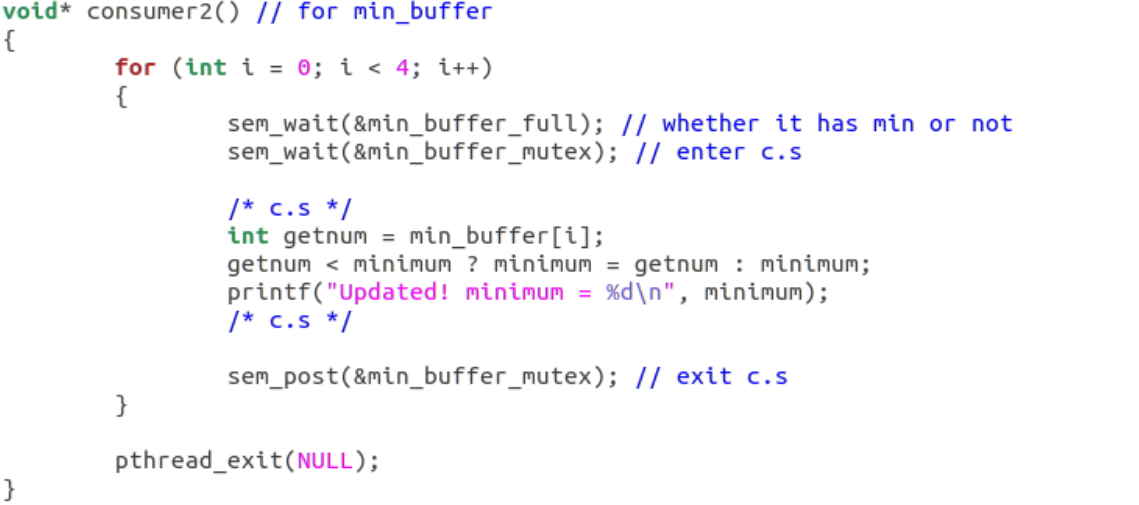
****

r1和r2代表thread所要處理的範圍(例如第一個是0~255)，會利用for迴圈找出範圍內的最大值和最小值。之後會將其放入buffer中，其中，因為buffer是shared data，所以我們會用semaphore來做存取控制，進入前會做wait而出來後會做post。

此外，consumer可能會因為目前buffer中沒有值可以取用而被block住，所以當我們在buffer中放值後會做post(若有被block住的thread即會被喚起)

1. **consumer function(1 and 2)**

****

****

在consumer function中，會依序去取出在buffer內的值和maximum /minimum比較，為避免存取到空的buffer，在進去之前會先利用wait function (則如果沒有資料就會被block住)

**功能實測**

