作業系統導論 LAB 2

系所：通訊三 學號 :409430030 姓名:翁佳煌

**<實驗工具>**

1.FreeRTOS即時作業系統。

2.JRE執行Eclipse的必備套件。

3.MinGW，Windows版的gcc。

4.Eclipse開發套件。

**<實驗過程與方法>**

下圖1，首先1~30行，都是一些參數宣告，特別注意的是，Mutex和Semaphore的宣告方式和用法。Mutex 只能被一個 Task 同時獲得，而 Semaphore 則可以被多個 Task 同時獲得。

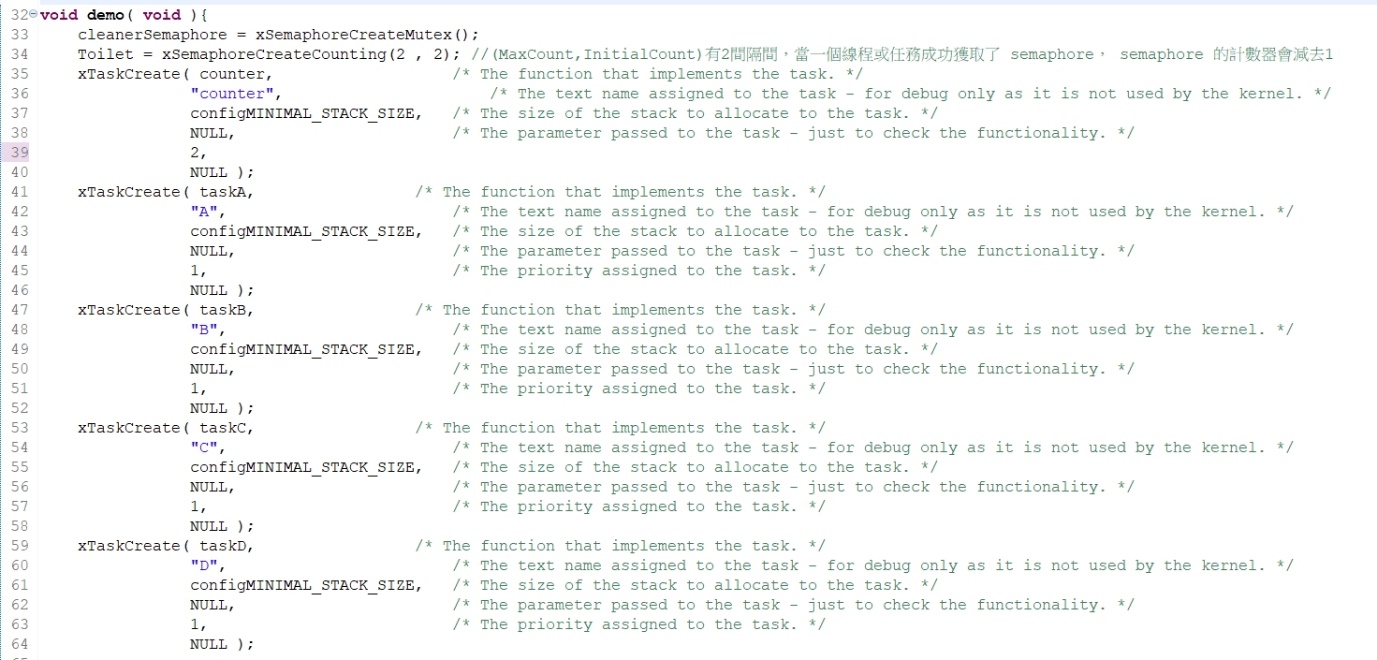
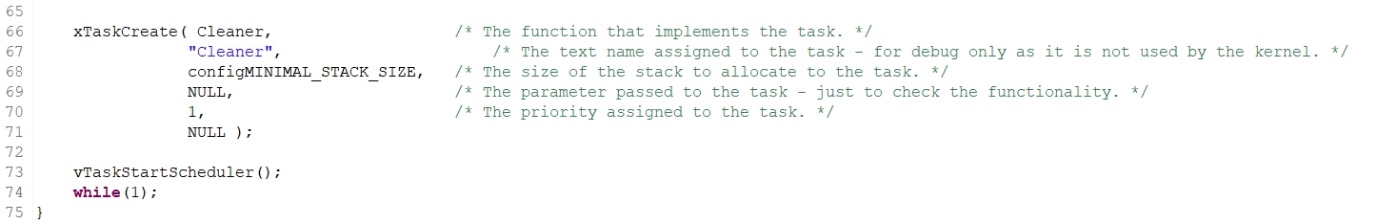


▲圖1

下圖2，第32~75行，首先值得注意的是，第33行的cleanerSemaphore，它是一個 mutex semaphore，也就是說只能有一個任務或線程能夠同時擁有它，其他任務或線程需要等待之前持有者釋放後才能獲得它。

而第34行的Toilet ，它是一個 counting semaphore，最初的計數器值是2，也就是說同時只能有2個任務或線程能夠獲得它，當一個線程或任務成功獲取了semaphore， semaphore 的計數器會減去1為建立任務。

接下來的幾行為題目要求，分別建立計算時間、4位學生和cleaner的task，其中，為了此實驗的目的，必須把4位學生和1為cleaner的權重設定為相同，在此我皆設定為1。



▲圖1.2

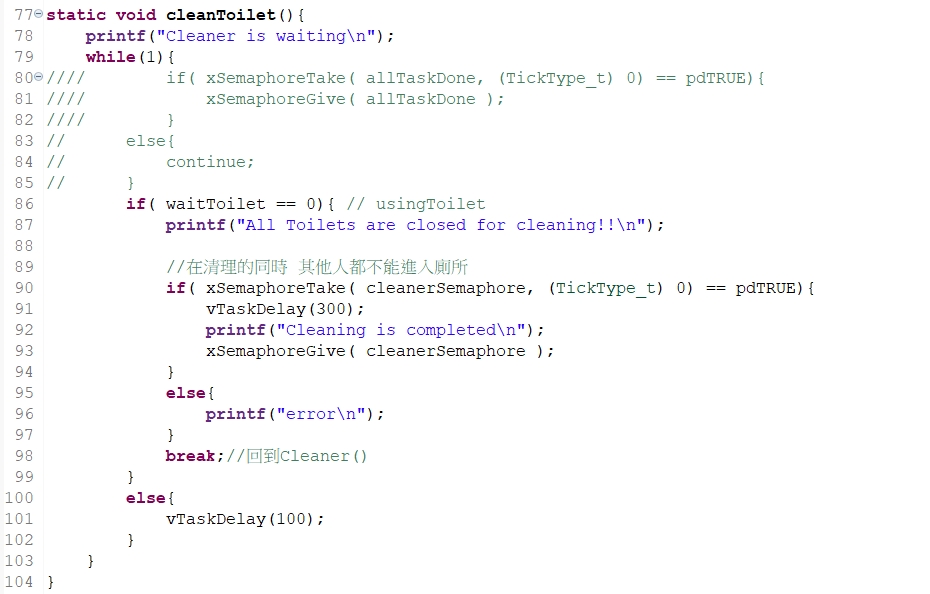
接下來77~104為cleanToilet函數，首先，程式會印出 "Cleaner is waiting" 這句話，表示清潔人員正在等待可以開始清潔廁所。

接下來，使用了一個 if 判斷式，判斷是否有使用者正在使用廁所。若 waitToilet 為 0，代表所有的廁所都關閉了，沒有人在使用，此時就可以開始進行清潔。

在清潔的同時，為了避免其他人進入廁所，使用了 xSemaphoreTake 函式來取得 cleanerSemaphore 這個 mutex。Mutex 是一個用於控制對共享資源存取的同步機制，當一個任務獲取了 mutex，其他任務就不能獲取，直到該 mutex 釋放。因此，當清潔人員正在清潔廁所時，其他使用者就不能進入廁所，以避免互相干擾。

接著，使用 vTaskDelay 函式延遲 300 個 tick，模擬清潔廁所的時間。當清潔完成後，使用 xSemaphoreGive 函式釋放 mutex，讓其他人可以進入廁所。

最後，使用 break 關鍵字跳出迴圈，回到 Cleaner() 函式。如果 waitToilet 不為 0，代表仍有人在使用廁所，此時就會等待 100 個 tick，再進行下一輪的檢查。



▲圖3

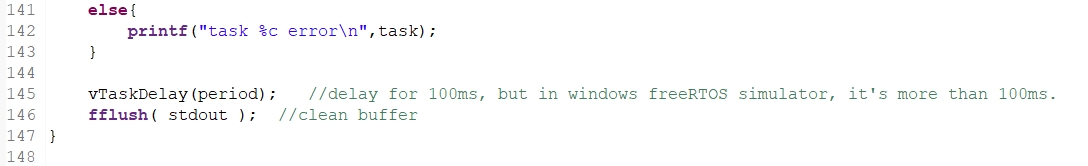
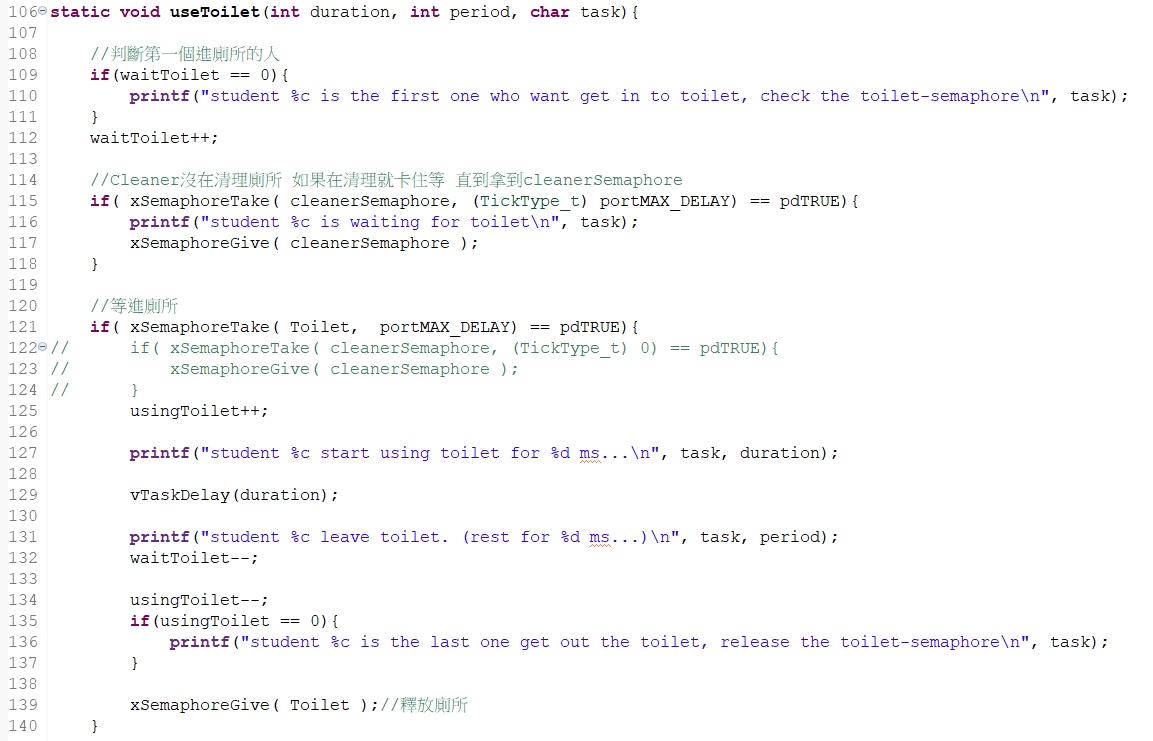
接下來106~148為useToilet函數，它有三個參數，分別是使用廁所的時間（duration）、離開廁所後的休息時間（period）、以及該學生的名稱（task）。

首先，函數會檢查這是否是第一個進入廁所的學生，如果是，就會印出一條訊息。

然後，函數會增加等待使用廁所的人數（waitToilet），代表有一個學生正在等待使用廁所。接著，函數會檢查清潔員是否在清理廁所，如果是，函數就會卡住等待，直到獲取到cleanerSemaphore，代表清潔員已經完成了清潔工作。

接下來，函數會等待進入廁所，使用xSemaphoreTake函數來獲取Toilet信號量。如果成功獲取到信號量，代表廁所可用，就會增加使用廁所的人數（usingToilet），然後印出使用廁所的訊息。之後，該學生會延遲duration毫秒，模擬使用廁所的時間。接著，該學生會印出離開廁所的訊息，並休息period毫秒。最後，函數會減少等待使用廁所的人數（waitToilet），以及減少使用廁所的人數（usingToilet）。如果這個學生是最後一個離開廁所的人，函數就會印出一條訊息，表示可以釋放Toilet信號量了。最後，函數會釋放Toilet信號量，以便其他學生可以進入使用。

最後，函數會延遲period毫秒，以便其他任務可以運行，並清空stdout緩衝區。

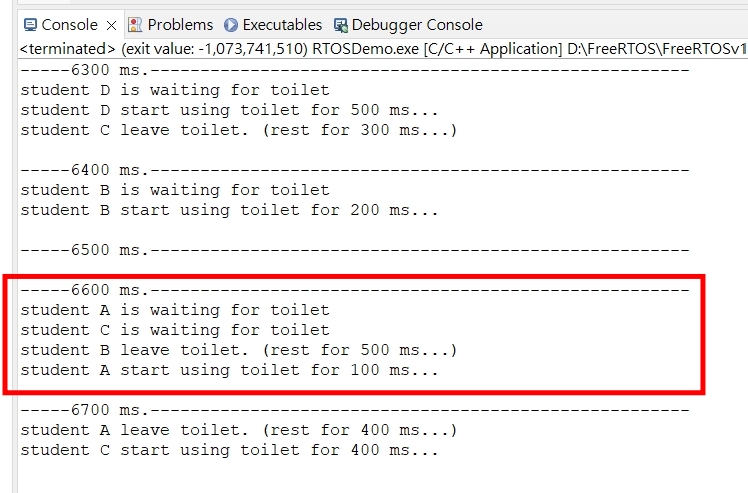


▲圖4

**<問題與討論>**

1. **在第 6600ms 時，誰在使用廁所?**

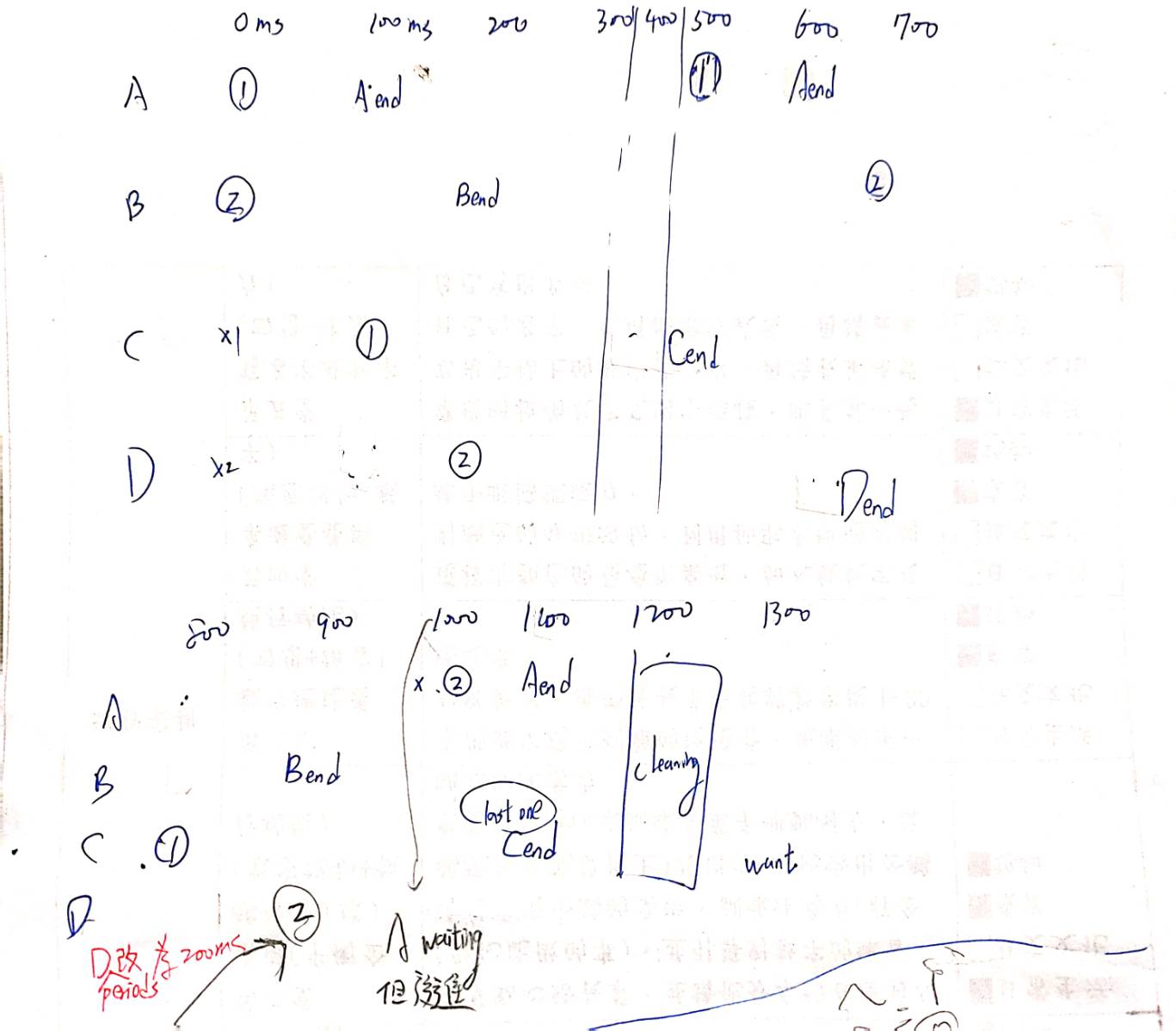
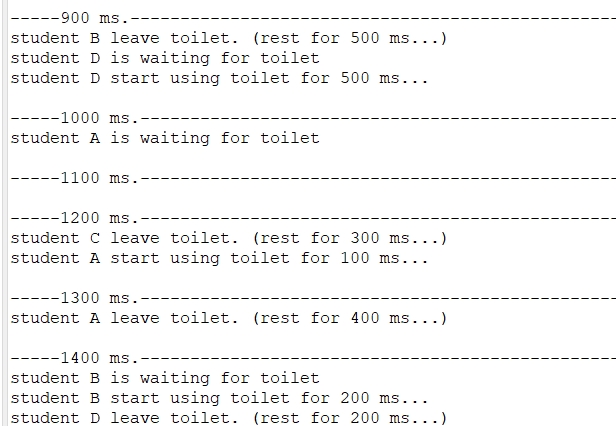
下圖5為執行結果，在第6600ms的時候A在使用廁所。



▲圖5

1. **當 Student D 的 period改為 200ms 時, 會發生什麼問題, 為什麼?**

下圖6為把Student D 的 period改為 200ms的執行結果，原本在第1200ms應該會全部人都離開使cleaner可以開始清理廁所，但這裡會發現如果把D改成200ms，會無法達成所有學生都不想上廁所的條件，導致cleaner無法清理廁所。



▲圖6

**<心得與收穫>**

這次是OS這堂課的第2個LAB，主要學到了如何使用FreeRTOS的mutex和counting semaphore來實現多線程的同步和互斥。mutex semaphore可以保證在線程中某些關鍵代碼區塊只能被一個線程訪問，從而防止競爭條件和數據不一致的問題。而counting semaphore則用於限制同一時間能夠訪問某些共享資源的線程數量，從而控制訪問頻率和訪問順序。在編寫多線程的code時，必須注意到競爭條件和數據不一致的問題，適當使用mutex和counting semaphore可以避免這些問題的發生。

這次的實驗花了我比較多時間，mutex和semaphore本身的概念並不難，往往是卡在邏輯上面的實現，花了很久搞清楚此時到底誰應該上廁所或是清理，我認為畫圖還是最快速可以搞懂的方法，總之，藉由這次LAB提高了我對多線程的概念和實作能力。

**<參考資料>**

[**http://wiki.csie.ncku.edu.tw/embedded/freertos**](http://wiki.csie.ncku.edu.tw/embedded/freertos)

<https://jasonblog.github.io/note/linux_system/mutex_yu_semaphore_zui_da_de_cha_yi_shi.html>

<https://ithelp.ithome.com.tw/articles/10279050>

<https://hackmd.io/@RinHizakura/rJhEpdyNw>