

## 6.1

假設今天帳戶餘額 200 元，丈夫要領 50 元，妻子要存 100 元，其正確的餘額應該會變成 250 元，但因為兩者過程會被序列化，可能會交錯排序，如：丈夫先領完了 50 元，妻子再存入 100 元，妻子完成存錢動作更新為 300 元，丈夫才更新餘額為 150 元，和預期出現出入。

此為 Critical section 問題，可以用 Peterson's Solution 解決，設兩個 process，共享兩個變數，turn 和 flag[2]當 A 在提款的時候，B 不能同時存款。

## 6.14

(a)Number\_of\_process 為會出現 race condition 的 variable，可能會同時被 allocate\_process()增加，release\_process()減少

(b)在進入每個 function 前調用 acquire()，在離開每一個 function 後立即調用 release()

## 6.33

見附檔

Define NR\_THREADS 為 threads 數(暫定為 10)