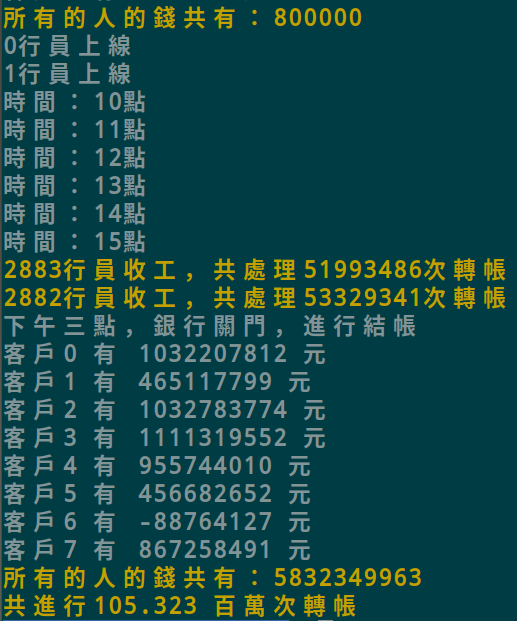
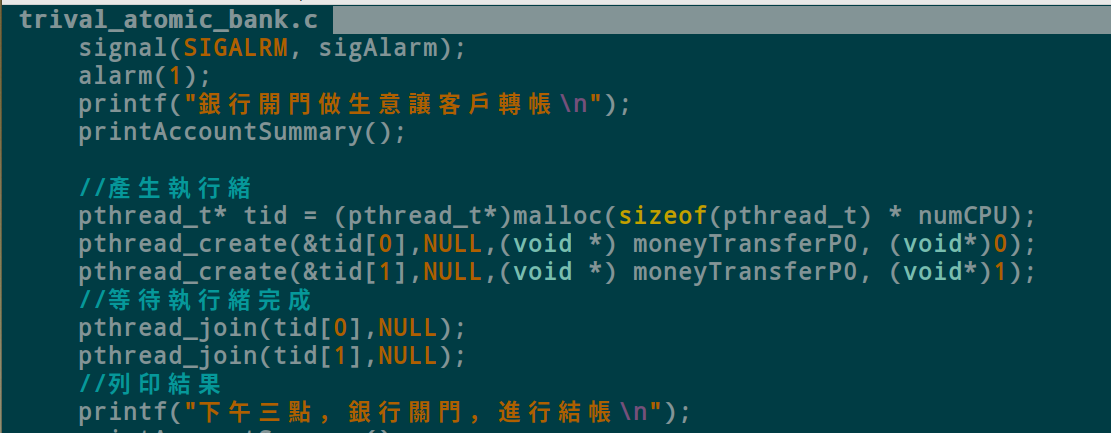
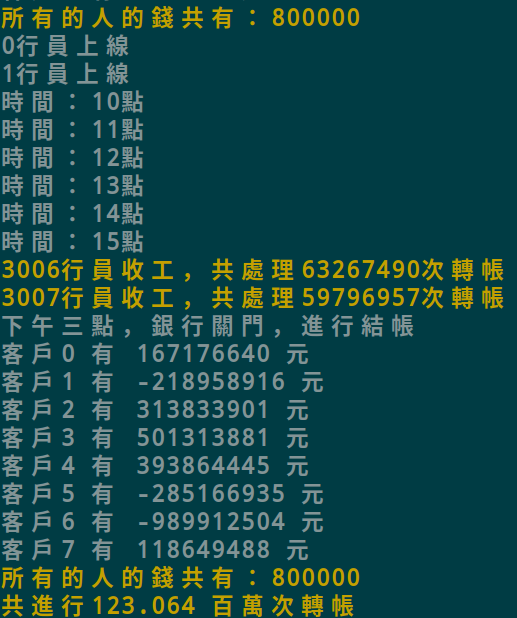
Ch05，番外篇一 synchronization

1. 找出對的那一種

(1)P0錯誤



(2)P1 正確



1. 然後簡單「說明」一下為什麼是對的
2. 因為P1使用了atomic\_fetch\_add和atomic\_fetch\_sub 可以保證不會互相競爭去改寫
3. 而P0使用了以下程式並且平行執行：

假設有兩個P0程式同時執行(為了方便辨識，修改了變數名稱，但程式碼架構不變)

行員一

long P1-tmp1 = bankAccount[source]-amount1;

long P1-tmp2 = bankAccount[dest1]+amount1;

atomic\_store(&bankAccount[source], P1-tmp1);

atomic\_store(&bankAccount[dest1], P1-tmp2);

行員二

long P2-tmp1 = bankAccount[source]-amount2;

long P2-tmp2 = bankAccount[dest2]+amount2;

atomic\_store(&bankAccount[source], P2-tmp1);

atomic\_store(&bankAccount[dest2], P2-tmp2);

(i)設兩支平行P0程式source剛好相同，

bankAccount[source] = 1000 ，bankAccount[dest1] = 3000，bankAccount[dest2] = 5000

amount1 = 500，amount2 = 1000

並且有可能用以下順序執行：

long P1-tmp1 = bankAccount[source]-amount1; // 500 = 1000-500

long P2-tmp1 = bankAccount[source]-amount2; // 0 = 1000-1000

long P2-tmp2 = bankAccount[dest2]+amount2; // 6000 = 5000+1000

atomic\_store(&bankAccount[source], P2-tmp1); // bankAccount[source] = 0

long P1-tmp2 = bankAccount[dest1]+amount1; // 3500 = 3000 +500

atomic\_store(&bankAccount[source], P1-tmp1); // bankAccount[source] = 500

atomic\_store(&bankAccount[dest2], P2-tmp2); // bankAccount[dest2] = 6000

atomic\_store(&bankAccount[dest1], P1-tmp2); // bankAccount[dest1] = 3500

故原本的bankAccount[source] + bankAccount[dest1]+ bankAccount[dest2]

= 1000+ 3000 +5000 = 9000

交易完後造成bankAccount[source] + bankAccount[dest1]+ bankAccount[dest2]

= 500 + 6000 +3500 = 10000

總金額不一致 故錯誤