

1.1. (15) ให้ count เป็น shared variable ที่มีค่าเริ่มต้นเป็นศูนย์ และเข้าถึงได้โดย thread 4 ตัว T1 T2 T3 และ T4 โดยแต่ละ thread มีการอัปเดตตัวแปร count ดังต่อไปนี้

T1: count++;

T2: count++;

T3: count--;

T4: count--;

เมื่อ thread ทั้ง 4 ทำงานเสร็จสิ้นลง ให้อธิบายว่าเป็นไปได้ไหมที่ค่า count จะมีค่าดังต่อไปนี้

- 0
- 2
- -1
- 3

กรณี 0 เป็นไปได้ ในกรณีที่ทุก thread รันแบบ atomic โดยไม่มีการเข้า race condition พร้อมกัน ก็คือไม่มีการแก้ค่า count พร้อมกัน ทำให้ผลลัพธ์ของ count เป็น 0 พอดี

กรณี 2 เป็นไปได้ ในกรณีที่ทุก thread รันพร้อมกันโดยไม่มีการกัน race condition ทำให้ มีโอกาสที่ T1 T2 รัน ทำให้ค่าเป็น 2 และอัปเดตค่าทีหลัง ทำให้เป็น 2

-> T1 count -> count++ -----> save ค่า  
-> T3 count -> count-- -> save ค่า -----  
-----> T4 รับค่า count -> count-- -> save ค่า

หลังจากนั้น T2 รับค่าล่าสุดที่ save จาก T1 และทำ count ++ และ save ค่า  
ทำให้ count = 2

กรณี -1 เป็นไปได้ คล้ายๆ กรณี count = 2 ที่ T3 หรือ T4 รับค่า count = 0 ก่อน แต่ว่า บันทึกราคาที่ได้หลังจากที่ Thread อื่นคิดค่าเสร็จแล้ว

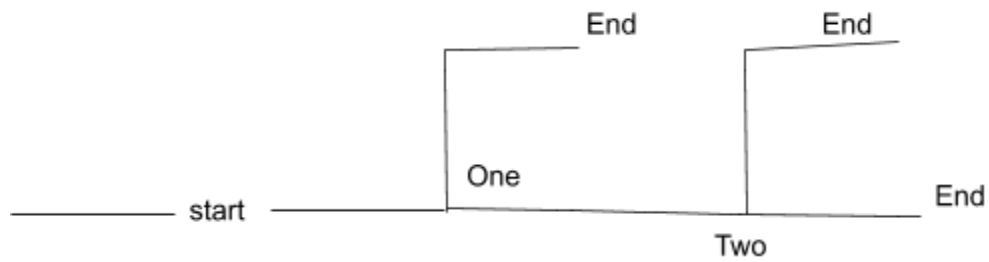
กรณี 3 เป็นไปไม่ได้ เนื่องจาก count บวกได้มากที่สุด จาก T1 T2 ที่ทำให้ค่ามากที่สุด = 2

## 1.2

```

void main()
{
    printf("Start\n");
    if (fork() != 0)
    {
        printf("One\n");
        if (fork() != 0)
        {
            printf("Two\n");
        }
    }
    sleep(1);
    printf("End\n");
}

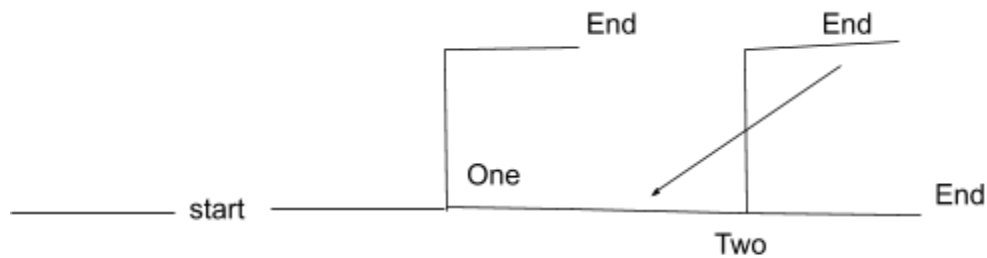
```



- เป็นไปได้ไหมที่ผลลัพธ์จะออกมาจะเป็นดังนี้

Start  
One  
End  
End  
Two  
End

เป็นไปได้ กรณี



Print start -> print One และ End และเมื่อ fork ครั้งที่ 2 print End ก่อน Two แล้วจบด้วยการ Print End ของ main

- และเป็นไปได้ไหมที่ผลลัพธ์จะออกมาจะเป็นดังนี้

Start  
End  
One  
End  
End  
Two

เป็นไปได้ เนื่องจาก End จะอยู่ท้ายสุดเสมอ จาก main แต่ End ระหว่างทางจพสามารถ สลับกับ One , Two ได้