

Figure 5.9: An Incorrect Solution in the Infinite-Buffer Producer/Consumer Problem Using Binary Semaphores

จากรูป 5.9 มีข้อผิดพลาดเกิดขึ้นตามลำดับเหตุการณ์ คือ

1. producer นำ item แรกไปใส่ใน buffer จะทำให้เงื่อนไข ($n==1$) เป็นจริง แล้ว producer จะ signal ค่า delay ให้มีค่าเป็น 1
2. consumer ทำงานจะไม่ block บน delay (ก่อนลูป while) และสมมติให้ทำมาจนถึงบรรทัด $n--$ ซึ่งทำให้ n มีค่าเป็น 0
3. producer นำ item ถัดไปใส่ใน buffer และพบว่าเงื่อนไข ($n==1$) เป็นจริง จึง signal ค่า delay เป็น 1 อีกครั้ง
4. consumer พบว่าเงื่อนไข $n==0$ เป็นเท็จ แล้ววนลูปไปถึง item ออกจาก buffer พร้อมลดค่า n ทำให้มีค่าเป็น 0 ถัดจากนี้ consumer จะพบว่าเงื่อนไข $n==0$ เป็นจริง และไม่ block บน delay (delay มีค่าเป็น 1 อยู่) ส่งผลให้เมื่อวนลูปรอบถัดไป จะไปลดค่า n เป็น -1 ซึ่งเป็นไปไม่ได้

ข้อผิดพลาดนี้เกิดขึ้นเนื่องจากการ signal ค่า delay ของ producer ในข้อ 3 ไม่ส่งผลต่อ consumer ในข้อ 4 ในจังหวะแรกทันที เพราะ consumer พบว่าเงื่อนไข $n==0$ เป็นเท็จ (consumer เห็น n เป็น 1 ขณะที่ producer เข้าใจว่า consumer น่าจะ block อยู่) และเมื่อ consumer ทำงานเร็วกว่า producer ทำให้ในที่สุด consumer ไปดึง item พร้อมลดค่า n จนทำให้เงื่อนไข $n==0$ เป็นจริง และพึงจะไม่ block บน delay ตามที่ producer เข้าใจก่อนหน้านี้ และทำงานต่อจนเกิดค่า n ที่เป็นไปไม่ได้

ข้อผิดพลาดนี้ไม่สามารถแก้ไขได้ด้วยการย้ายคำสั่ง $if (n==0)$ ของ consumer ไปไว้ก่อน signal ค่า s เนื่องจากการกระทำได้กล่าวทำให้เกิดผล ดังนี้

1. producer นำ item แรกไปใส่ใน buffer จะทำให้เงื่อนไข ($n==1$) เป็นจริง แล้ว producer จะ signal ค่า delay ให้มีค่าเป็น 1
2. consumer ทำงานจะไม่ block บน delay (ก่อนลูป while) และสมมติให้ทำมาจนถึงบรรทัด $n--$ ซึ่งทำให้ n มีค่าเป็น 0 และทำงานต่อเนื่องไปจนถึงพบว่าเงื่อนไข $n==0$ เป็นจริง และ block อยู่บน delay จึงไม่สามารถ signal ค่า s ในขั้นตอนถัดไปได้ ส่งผลให้ producer ต้อง block อยู่บน $s ->$ ทั้ง producer และ consumer ต่าง block อยู่บน semaphore คนละตัวกัน ไม่มีใครจะไป unblock อีกฝ่ายได้ เรียกได้ว่า เกิด deadlock ขึ้น

ข้อผิดพลาดนี้สามารถแก้ไขได้ดังรูป 5.10 ด้วยการสร้างตัวแปร m ในโค้ดของ consumer เพื่อเก็บค่า n เดิมก่อนออกจาก critical section ไว้ แล้วค่อยนำมาทดสอบภายหลังแทนการทดสอบตัวแปร n ที่อาจถูก producer เปลี่ยนแปลงค่าอย่างในกรณีที่เกิดข้อผิดพลาดที่ผ่านมา