

Case Study 2

โปรแกรมจากโจทย์ทำงานไม่ได้ตามที่ต้องการเพราะเหตุใด

จากโปรแกรมเป็นการทำงานแบบ multi-thread แต่รูปแบบการเขียนโปรแกรมไม่ได้ถูกออกแบบมาเพื่อรองรับการทำงานที่เป็นแบบ multi-thread ซึ่งจะทำให้เกิดปัญหา race condition ขึ้น ดังนั้นจึงทำให้การทำงานของโปรแกรมออกมาไม่ถูกต้อง และโปรแกรมไม่มีคุณสมบัติ Thread safe หรือก็คือการที่โปรแกรมสามารถรองรับการทำงานแบบ multi-thread และสามารถยังทำงานได้อย่างถูกต้องตามตรงการ

มีจุดใดบ้างที่ต้องแก้ไขอธิบายพร้อมเหตุผลรองรับ

ภายในฟังก์ชันการทำงานของแต่ละ thread สำหรับหน้าที่ enqueue และ dequeue ต้องมีการเขียนโปรแกรมที่รองรับการทำงานแบบ Synchronization เพื่อเป็นการป้องกันปัญหา race condition ที่เกิดขึ้นจากการทำงานแบบ multi-thread โดยจะมีการประยุกต์ใช้หลักแนวคิดของทั้ง Lock, Semaphore และ Condition Variables ในการช่วยให้การทำงานของโปรแกรมออกมาได้อย่างถูกต้องตามต้องการ

สรุปผลการทดลอง

Version 1

ผลลัพธ์

โปรแกรมสามารถทำงานแบบ multi-thread แบบ 2 thread ได้แล้วคือ thread สำหรับ enqueue กับ thread สำหรับ dequeue โดยสามารถได้ผลลัพธ์ออกมาได้อย่างเรียงลำดับ 1-50 อย่างถูกต้อง

วิธีการแก้ปัญหา

มีการประยุกต์ใช้ Semaphore มาเพื่อให้โปรแกรมมีการทำงานเป็นแบบ Synchronous ระหว่างทั้งสอง thread เพื่อที่จะให้การ dequeue ออกมาเป็นลำดับตามตัวที่ enqueue เข้ามาภายใน buffer

ปัญหาที่พบใน version นี้

โปรแกรมยังไม่มีคุณสมบัติ Thread safe หมายความว่าถ้าในกรณีที่ยากเพิ่ม thread สำหรับหน้าที่ dequeue ก็จะทำให้โปรแกรมไม่สามารถทำงานได้อย่างถูกต้องตามที่ต้องการได้

Version 2

ผลลัพธ์

โปรแกรมสามารถทำงานแบบ multi-thread ได้ซึ่งจะมี 2 thread ที่ทำหน้าที่ dequeue ซึ่งจะให้ผลลัพธ์ออกมาได้ ซึ่งผลลัพธ์ที่ได้ออกมาก็จะเรียงลำดับ 1-50 และมีการสลับการทำงานของ thread ในการ dequeue กันไประหว่างทั้งสอง thread

วิธีการแก้ปัญหา

กำหนดให้ฟังก์ชันสำหรับการ enqueue สามารถ enqueue ได้เพียง 10 element ต่อหนึ่งครั้งการทำงานของ thread นั้นเท่านั้น ซึ่งมาจากการประยุกต์ให้โปรแกรมทำงานแบบ Synchronization ด้วย Semaphore ให้ thread ที่ทำการ dequeue ทั้งสอง thread มารอคิวในการทำงานโดยให้แต่ละ thread dequeue เป็นจำนวน 5 ครั้ง ซึ่งมาจากจำนวน element ทั้งหมดภายใน buffer หารด้วยจำนวน thread

ปัญหาที่พบใน version นี้

โปรแกรมยังไม่สามารถทำงานได้แบบ Thread safe คือถ้าเพิ่ม thread ที่ทำหน้าที่ในการ dequeue มา โปรแกรมก็จะให้ผลลัพธ์ที่ผิดมา

Version 3

ผลลัพธ์

มีการทำงานของโปรแกรมแบบ Thread safe คือโปรแกรมยังสามารถทำงานได้ผลลัพธ์เหมือนเดิม ถึงแม้ว่าจะมีการเพิ่มจำนวนการทำงานของ thread ที่ทำงานพร้อมกันมากขึ้น และยังมีการให้ผลลัพธ์การทำงานได้อย่างถูกต้องตามต้องการ

วิธีการแก้ปัญหา

เริ่มต้นด้วยการกำหนดหน้าที่และจังหวะการทำงานของแต่ละ thread ก่อนคือจะมี thread ที่ทำหน้าที่ enqueue คือจะเริ่มการทำงานก่อน โดยจะให้ thread ที่ทำหน้าที่ enqueue คือจะเริ่มการทำงานก่อน โดยจะให้ enqueue ไปทั้งหมด 10 element (ตามจำนวนขนาดของ buffer) แล้วจากนั้นจะให้ thread ที่ทำหน้าที่ dequeue ทำงานจนกว่าจะหมดหน้าที่ จากนั้น thread ที่ทำหน้าที่ enqueue ก็สามารถทำงานต่อได้ แล้วทำงานวนลูปไปเรื่อยๆ จนกว่าจะสามารถ enqueue ทั้งหมด 50 element ครบตามต้องการ

ปัญหาที่พบใน version นี้

โปรแกรมสามารถทำงานแบบ thread safe ได้แล้ว แต่ยังมีปัญหาในกรณีที่มีการเพิ่ม thread ที่ทำงาน enqueue ขึ้นมาก็จะทำให้โปรแกรมยังไม่สามารถทำงานได้

Version 4

ผลลัพธ์

สามารถแก้ปัญหาจาก version ก่อนหน้าได้แล้ว ซึ่งตัวโปรแกรม ซึ่งตัวโปรแกรมสามารถทำงานสำหรับ thread ที่ enqueue 1-50 เสร็จก่อนแล้วจากนั้นให้ทำงาน thread ที่ enqueue 101-150 ต่อได้ ส่วนการ dequeue ก็สามารถทำออกมาได้ผลลัพธ์อย่างถูกต้องเหมือนเดิม

วิธีการแก้ปัญหา

เป็นการเพิ่ม Conditional Variables ไปสำหรับตอนที่ทำงาน enqueue ของ 1-50 สำเร็จแล้ว จะให้ thread ที่ enqueue 101-150 สามารถเริ่มการทำงานต่อได้

ปัญหาที่พบใน version นี้

การทำงานของโปรแกรมยังไม่สามารถนับว่าเป็นการทำงานแบบ Thread safe ได้ 100% เพราะว่าถ้าเกิดมีการเพิ่ม thread ที่ทำหน้าที่ enqueue เข้าไป ก็จำเป็นที่จะต้องแก้ไขโค้ดเพิ่มเติมในส่วนอื่นอีกด้วย

Version 5

ผลลัพธ์

โปรแกรมสามารถทำงานได้อย่างถูกต้อง และมีการทำงานแบบ thread safe ได้

วิธีการแก้ปัญหา

เปลี่ยนจากที่การประยุกต์การทำงานโปรแกรมให้เป็น Synchronization ด้วย Semaphore ให้มาใช้ Lock แทน เพราะว่าจะมีลักษณะการทำงานระหว่าง thread จะเป็นการทำงานแบบสุ่ม ไม่มีลำดับการทำงานของ thread ทำให้จำเป็นต้องมีการเขียน Conditional Variables ดักไว้เพื่อไม่ให้เกิดการทำงานที่ผิดพลาด อย่างเช่นในการทำงานของ thread enqueue จะไม่สามารถทำงานได้เมื่อใน buffer มี element ที่ยังไม่ถูก dequeue เต็มตามขนาดของ buffer ซึ่งคือ 10

สมาชิก

63010035	กฤษฎาง	แสงศิริผล
63010082	กิตติภณ	สิงห์ชม
63010183	ชนน	กุลกัต์ติมาส
63010229	ชินกฤต	ปิ่นคล้าย
63010283	ณภัทร	มูลพินิจ
63011013	สุเมธ	สวนสำราญ