****

**รายงาน การออกแบบ**

**จัดทำโดย**

นางสาวกนกวรรณ บัวภาคำ 6040200049

นายจารุเดช ก่อรักเสวต 6040200618

นายธรรมธัช ตันติปิธรรม 6040201983

นางสาวธัญลักษณ์ โภคธนาพิพัฒนพร 6040202033

นายธีระวัฒน์ ชรินทร์ 6040202203

นายนฤปราชญ์ ศักดิ์ชัยพานิชกุล 6040202343

นายพีรพล วรรณพันธ์ 6040203412

นางสาวเพ็ญรดี โพนเมืองหล้า 6040203463

นายภาคภูมิ วาดวงศ์ 6040203609

นางสาวรัดดาพร อักษรทอง 6040204010

นายอนุชา ศรีลาแก้ว 6040205407

นายอำนาจ ทนงนวล 6040205938  
  
**มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร**

**ภาคต้น ปีการศึกษา 2562**

**คำนำ**

รายงานการออกแบบวิชาระบบปฏิบัติการคอมพิวเตอร์ รหัสวิชา ๐๑๒๐๔๓๓๒ หมู่เรียน ๑ มีจุดประสงค์เพื่อรายงานผลการวิเคราะห์การออกแบบ ซึ่งรายงานฉบับนี้มีเนื้อหาเกี่ยวกับการออกแบบ เงื่อนไข วิธีการทำงานของฟังก์ชันต่างๆ อธิบายความถูกต้องของโปรแกรม ผลลัพธ์ของการรันโปรแกรม ซอร์สโค้ดของโปรแกรม

คณะผู้จัดทำได้ศึกษาค้นคว้าข้อมูลของบัฟเฟอร์แบบวนกลับ เทรด การเขียนฟังก์ชัน append remove และ buff ตามโจทย์ที่อาจารย์ผู้สอนได้ให้มา จากแหล่งข้อมูลต่างๆ

การจัดทำรายงานฉบับนี้สำเร็จตามวัตถุประสงค์ไปด้วยดี ทั้งนี้ทางคณะผู้จัดทำได้หวังเป็นอย่างยิ่งว่าการศึกษาค้นคว้าข้อมูลจะมีประโยชน์ต่อผู้ที่ได้มาศึกษาต่อเป็นอย่างดี หากมีสิ่งที่ต้องปรับปรุงแก้ไขเพิ่มเติมประการใด คณะผู้จัดทำน้อมรับฟังคำแนะนำ ข้อเสนอแนะที่มีต่อเนื้อหาเพื่อนำไปปรับปรุงเอกสารรายงานให้สมบูรณ์ยิ่งขึ้น

๒ พฤศจิกายน ๒๕๖๒

คณะผู้จัดทำ

**สารบัญ**

**เนื้อหา หน้า**

คำนำ……………………………………………………………………………………….... I

สารบัญ…………………………………………………………………………………….... II

สารบัญภาพ…………………………………………………………………………………. III

1. การออกแบบโปรแกรม…………………………………………………………………...

1.1 การออกแบบ Buffer…………………………………………………………………..

1.2 การออกแบบ Append………………………………………………………………....

1.3 การออกแบบ Remove…………………………………………………………………

1.4 Flowchart ของโปรแกรม………………………………………………………………

2. เงื่อนไข วิธีการทำงาน และการพิสูจน์คุณสมบัติของ Append……………………………..

2.1 เงื่อนไขของ Append……………………………………………………………………

2.2 วิธีการทำงานของ Append………………………………………………………………

2.3 การพิสูจน์คุณสมบัติของ Append……………………………………………………….

3. เงื่อนไข และวิธีการทำงานของ Remove…………………………………………………….

3.1 เงื่อนไขของ Remove…………………………………………………………………….

3.2 วิธีการทำงานของ Remove……………………………………………………………....

4. ผลการ Run & Result………………………………………………………………………...

5. ซอร์สโค้ดของโปรแกรม…………………………………………………………………….

เอกสารอ้างอิง…………………………………………………………………………………..

**สารบัญภาพ**

**เนื้อหา หน้า**

รูปที่ 1 อธิบายการทำงานของโปรแกรม 7

รูปที่ 2 Flowchart อธิบายการทำงานของ Append 8

1. **การออกแบบโปรแกรม**

**1.1 การออกแบบ Buffer**

Buffer ที่ออกแบบเป็นประเภท Circular Buffer โดย Implement เป็น Array of Boolean โดยแรกเริ่ม Buffer ตั้งต้นมีค่าเป็น 0 (No Item) และมี Integer 2 จำนวน เสมือนเป็น Pointer ชี้ตำแหน่ง Head กับ Tail ทั้งสองจะถูกกำหนดให้อยู่ในตำแหน่งเดียวกัน คือ Index ที่ 0 เมื่อมีการ Add Item จะเปลี่ยนค่า Buffer ในตำแหน่งที่ Tail ชี้อยู่เป็น 1 และขยับตำแหน่งของ Tail ไปในตำแหน่งถัดไป ในทำนองเดียวกัน เมื่อมีการ Remove Item จะเปลี่ยนค่า Buffer ในตำแหน่งที่ Head ขึ้นอยู่เป็น 0 และขยับตำแหน่งของ Head ไปในตำแหน่งถัดไปเช่นกัน เมื่อ Head หรือ Tail ถึงตำแหน่งสุดท้ายของ Array ในการประมวลผลครั้งต่อไปจะถูกขยับกลับมาเริ่มใหม่ในตำแหน่งแรกเริ่ม

มี Producer และ Consumer ซึ่งเป็น Thread ที่ใช้งาน Buffer โดย Producet จะเรียกใช้งาน Function Append ที่เพิ่มข้อมูลลงใน Buffer ในตำแหน่งที่ Tail นี้อยู่ และ Consumer จะเรียกใช้งาน Function Remove ที่ลบข้อมูลออกจาก Buffer ในตำแหน่งที่ Head ชี้อยู่

Buffer มีการวัดประสิทธิภาพการทำงานโดยการจับเวลาตั้งแต่เริ่มทำงานจนจบการทำงาน มีการนับจำนวน Request ทั้งหมดที่ทำงานสำเร็จแล้วคำนวณออกมาเป็นเปอร์เซนต์และอัตราการทำงานต่อวินาทีได้

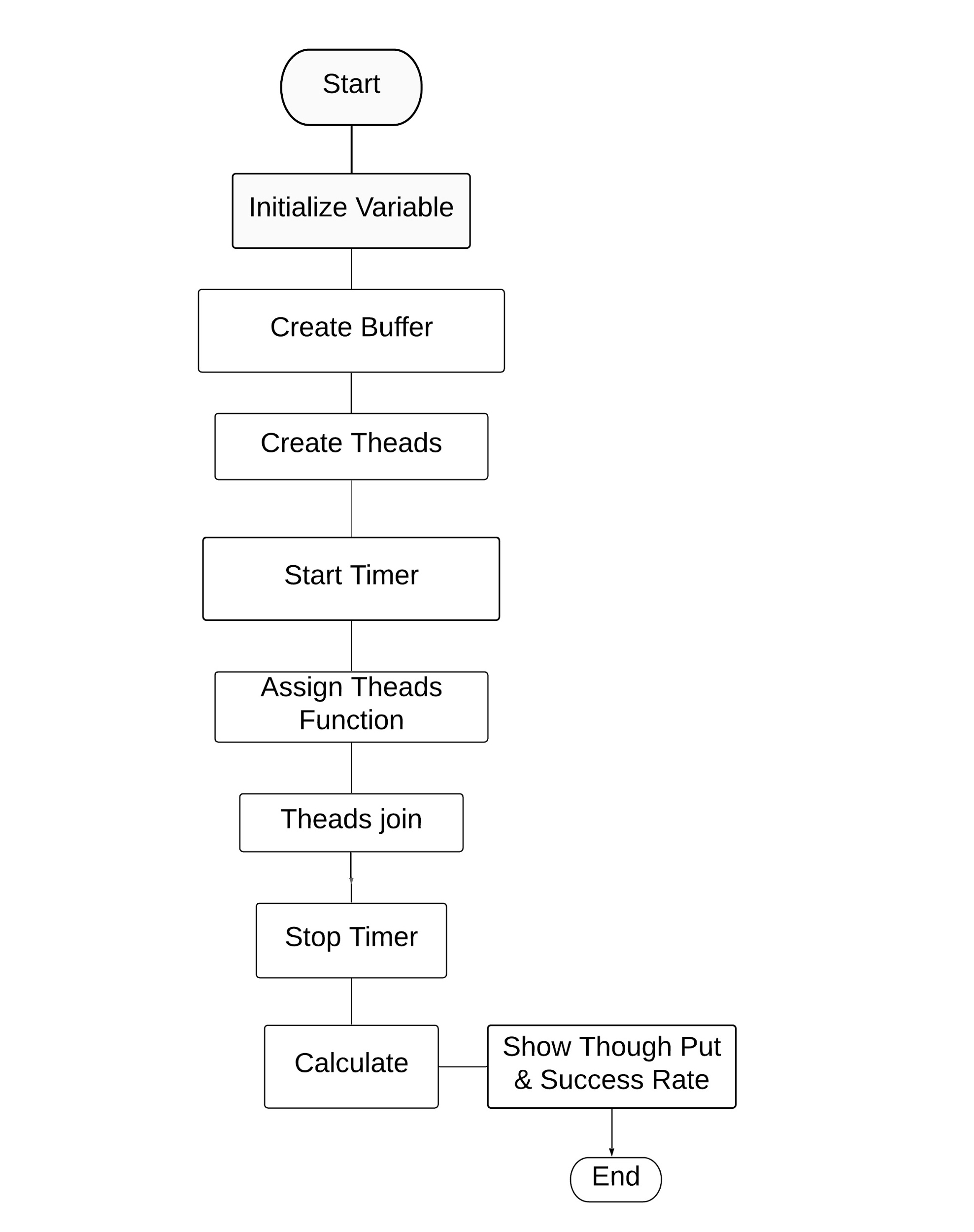
**1.2 การออกแบบ Append**

Append ทำหน้าที่ Add item ลงใน Buffer กรณีที่ Buffer เต็มจะให้ Reqest ที่เข้ามารอจนกว่า Buffer จะว่าง หรือ TimeOut และเนื่องจาก Producer อาจจะไม่ได้มีแค่ 1 Thread เพื่อป้องกันการถูกทับซ้อนกันของ Buffer ดังนั้นเมื่อมีการเข้าสู่การทำงานของ Append จะมีการ Lock ไว้ด้วย โดยมีการใช้คุณสมบัติของ Mutex จนกว่าฟังก์ชันของ Append จะจบการทำงานหรือจำนวน Reqest หมดจะ Unlock เพื่ออนุญาตให้ Thread อื่นสามารถเข้าใช้ Buffer ได้ต่อไป

**1.3 การออกแบบ Remove**

Remove ทำหน้าที่ Remove Item ออกจาก Buffer กรณีที่ Buffer ว่าง จะไม่เกิดการ Remove ขึ้น เมื่อเข้าสู่การทำงานของ Remove จะมีการล็อคไว้ด้วย โดยใช้คุณสมบัติของ Muter จนกว่าฟังก์ชัน Remove จบการทำงาน หรือ จำนวน Request หมด และ Buffer ว่าง

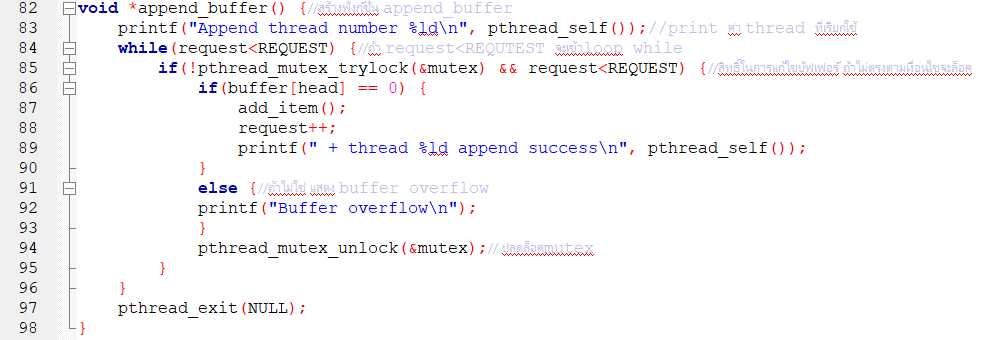
**1.4 Flowchart ของโปรแกรม**

****

**รูปภาพที่ 1 อธิบายการทำงานของโปรแกรม**

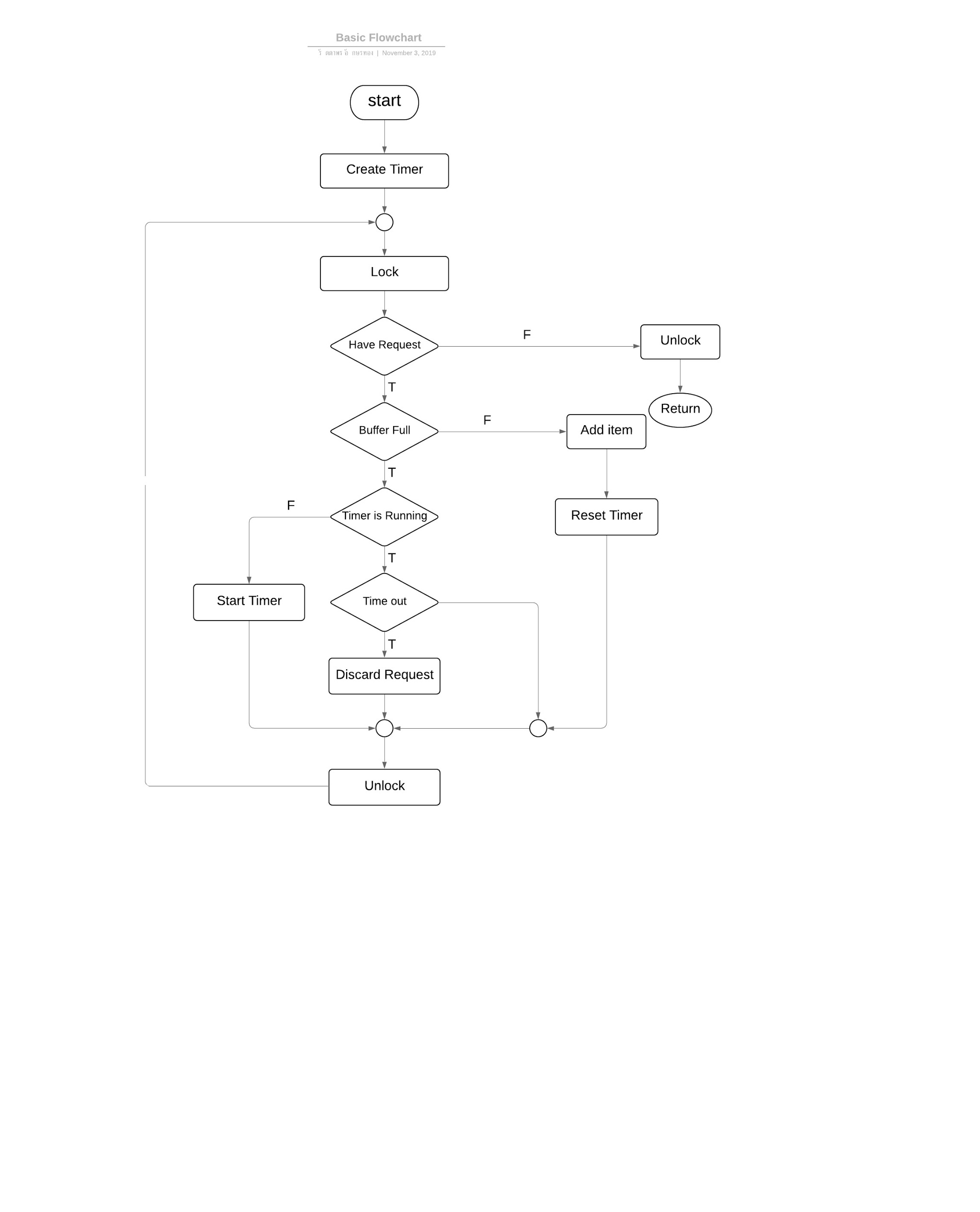
**2. เงื่อนไข วิธีการทำงาน และการพิสูจน์คุณสมบัติของ Append**

**2.1 เงื่อนไขของ Append**

****รูปภาพที่ 2 คือเงื่อนไขของ Append

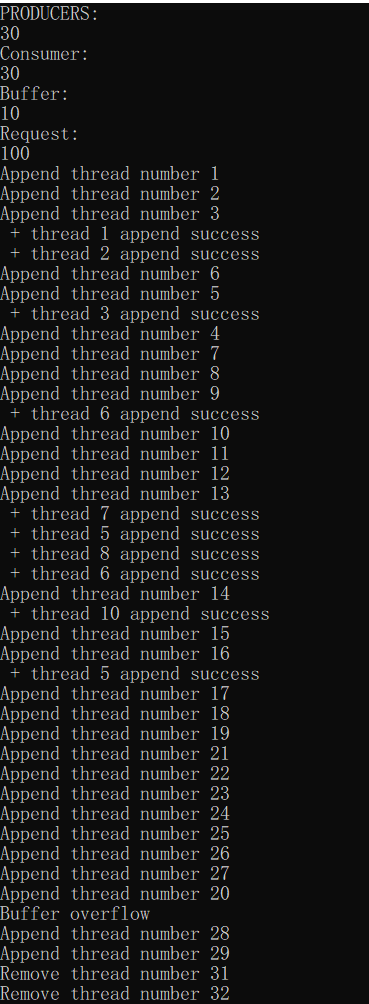
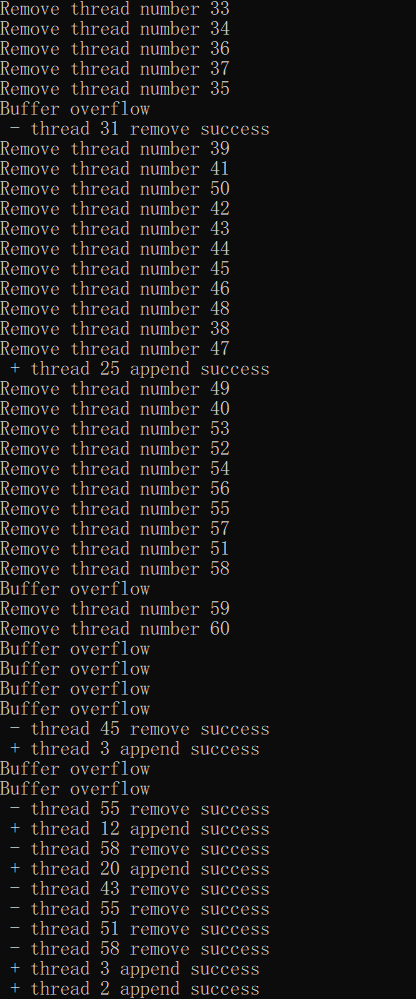
**2.2 วิธีการทำงานของ Append**

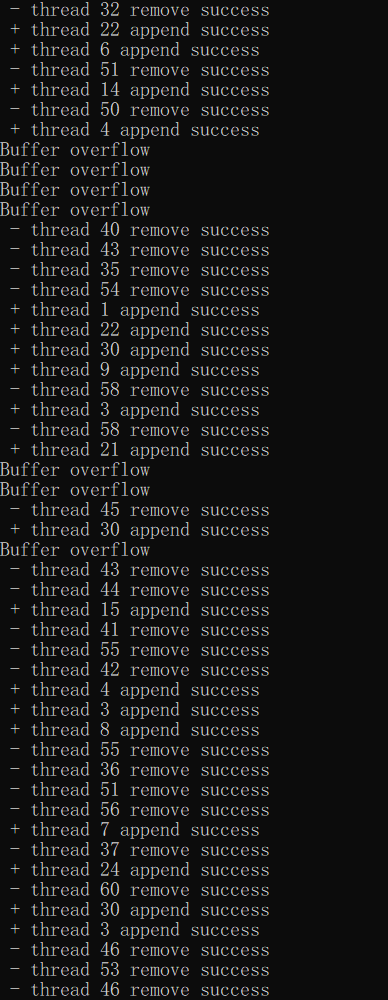
เมื่อทำการเลือกฟังก์ชัน Add\_item โปรแกรมจะเพิ่มรายการที่เรากรอกเข้าไปที่ท้ายแถวของบัฟเฟอร์โดยสามารถเพิ่มได้ N รายการเมื่อเพิ่มรายการจนถึง N รายการแล้วจะไม่สามารถเพิ่มรายการได้อีกแล้วโปรแกรมจะแจ้งว่า Buffer is overflow ซึ่งจะต้องรอให้บัฟเฟอร์มีพื้นที่ว่างก่อนถึงจะสามารถเรียกใช้ฟังก์ชัน Add\_item ได้อีก

****

**รูปภาพที่ 2 Flowchart อธิบายการทำงานของ Append**

**2.3 การพิสูจน์คุณสมบัติของ Append**

ผู้จัดทำพิสูจน์คุณสมบัติของ Append โดยการ กำหนดค่า Buffer และ Removeมีจำนวนน้อยลง กำหนดค่า Producer มีจำนวนมากขั้น ทั้งนี้เพื่อให้การ Run โปรแกรมมีโอกาสเกิด TimeOut จากจำนวนของ Buffer เต็มมากขึ้น แสดงลักษณะ ดังรูป** **

****

**3. เงื่อนไข และวิธีการทำงานของ Remove**

**3.1 เงื่อนไขของ Remove**

void \*remove\_item() {//สร้างฟังก์ชั่น remove\_item

buffer[tail++] = 0;//buffer[tail++]มีค่า=0

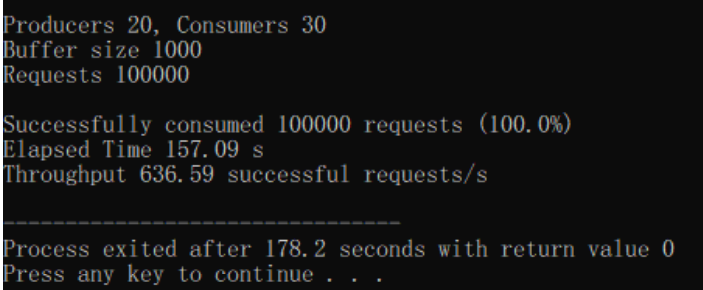
tail = tail % BUFFER\_SIZE;//tail มีค่าเท่ากับ tail / buffer\_size เอาแต่เศษ

**3.2 วิธีการทำงานของ Remove**

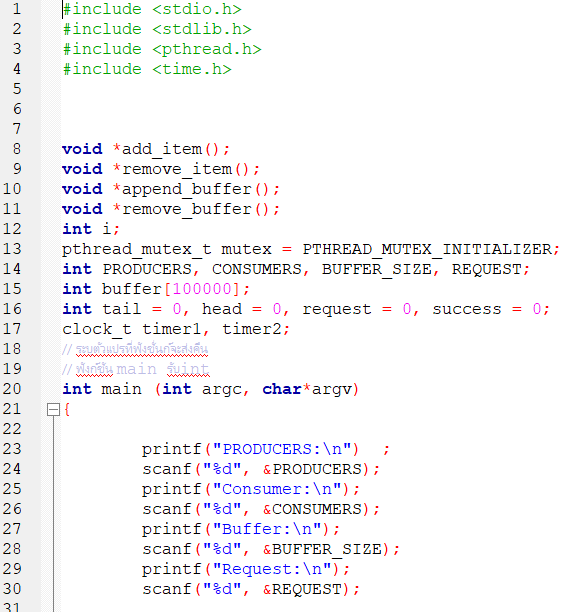
เมื่อทำการเลือกฟังก์ชัน Remove\_item โปรแกรมจะลบรายการในบัฟเฟอร์ออก โดยจะลบรายการที่อยู่หน้าแถวออกก่อน กล่าวคือ โปรแกรมจะทำการลบรายการที่ถูกเพิ่มเข้ามาในบัฟเฟอร์ก่อน และถ้าลบข้อมูลในบัฟเฟอร์จนหมดแล้วโปรแกรมจะแจ้งว่า No buffer underflow คือไม่มีรายการเหลืออยู่ในบัฟเฟอร์แล้ว และจะต้องรอให้บัฟเฟอร์ไม่ว่างถึงจะเรียกฟังก์ชัน Remove\_item ได้

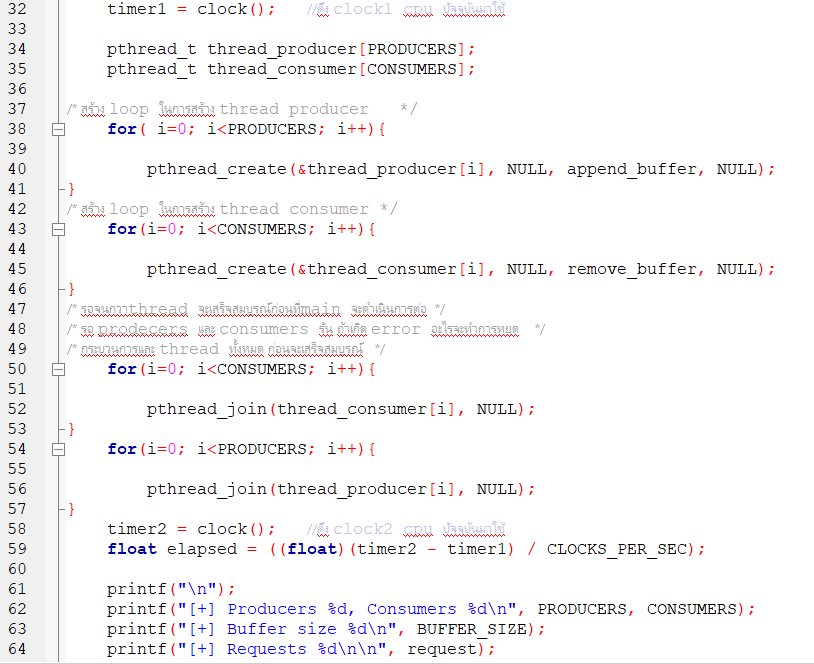
**4. ผลการ Run & Result**

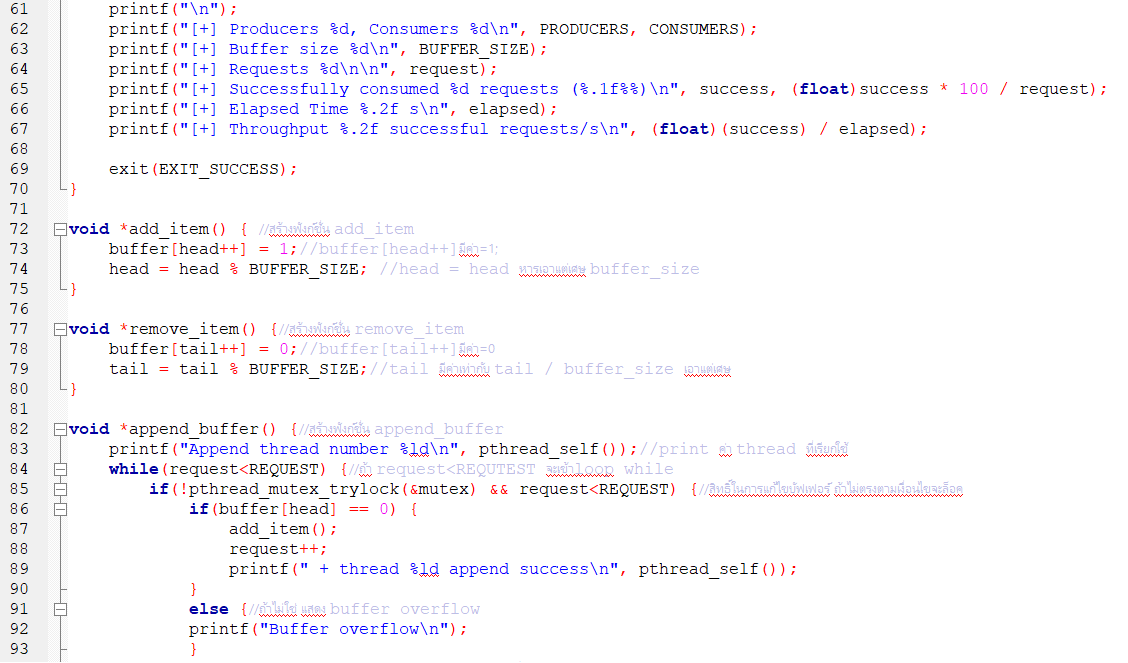
สามารถอธิบายผลการ Run & Result ได้ดังรูปที่

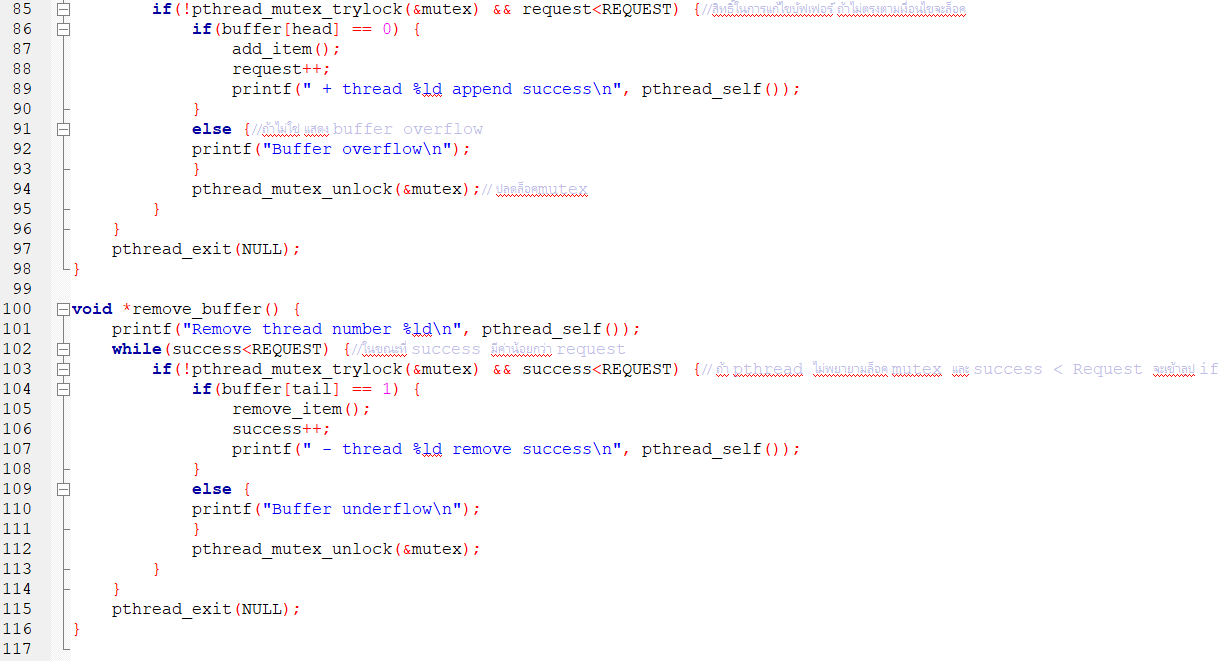


**5. ซอร์สโค้ดของโปรแกรม**

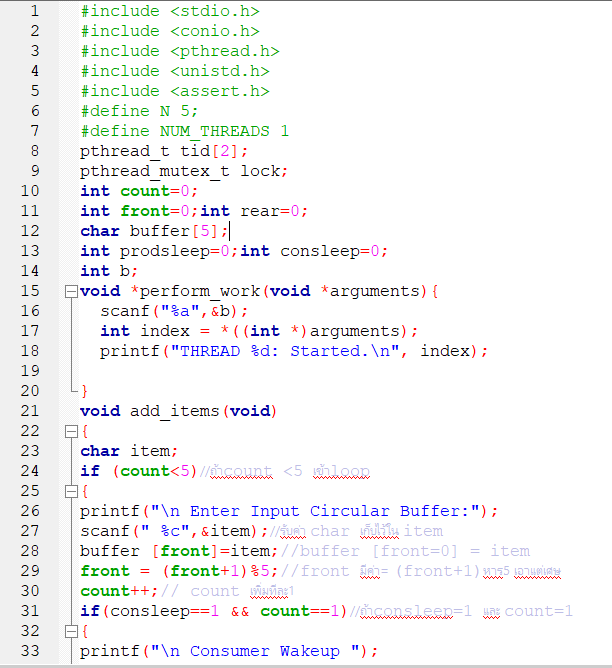


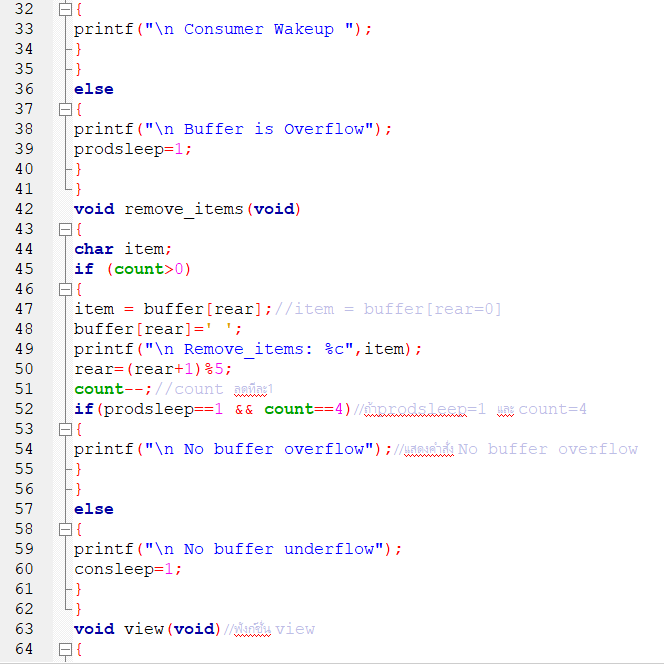


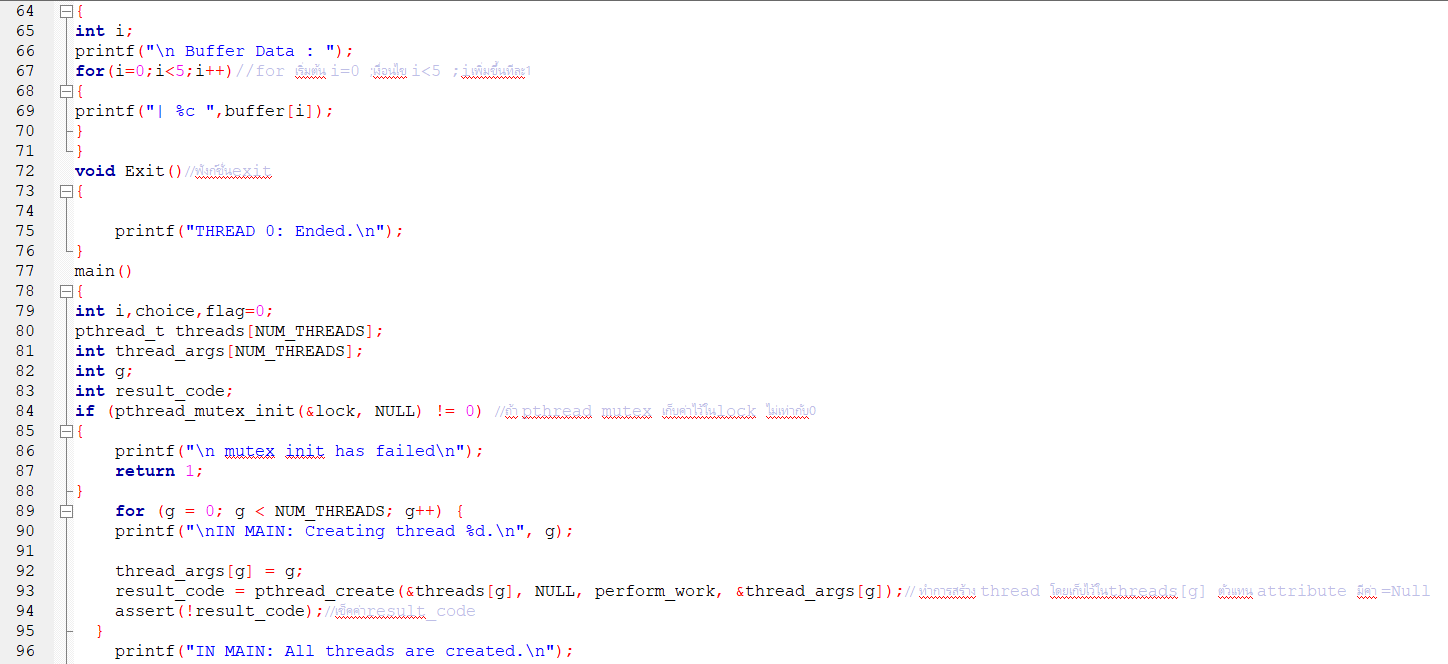


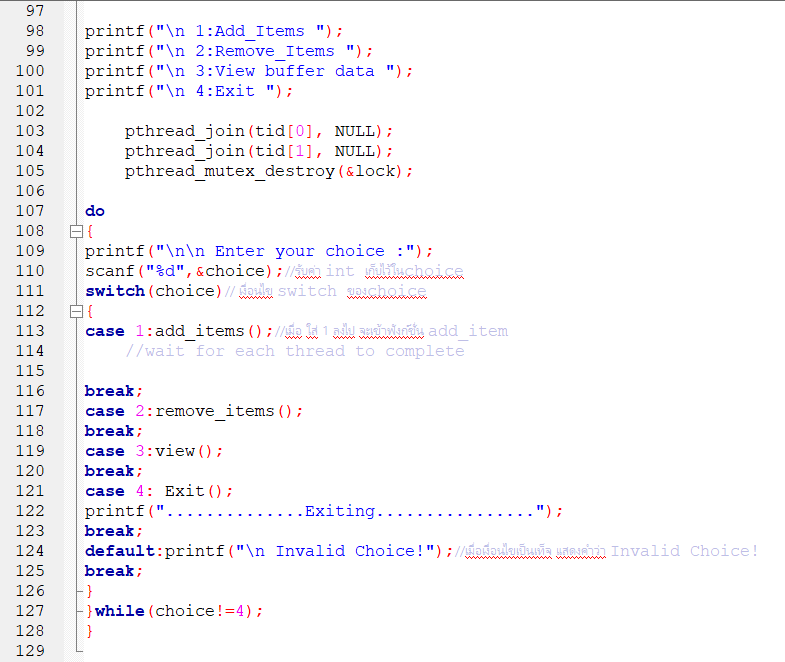


-----------------------------------------------------------------------------------------------------------------

****

****

****

****

**เอกสารอ้างอิง**

**Producer Consumer Problem without using semaphore** (ออนไลน์) แหล่งที่มา : <http://babulax.blogspot.com/2010/04/producer-consumer-problem-without-using.html>

**POSIX thread libraries** (ออนไลน์) แหล่งที่มา :

<http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>

**Pthread with circular buffer** (ออนไลน์) แหล่งที่มา :

https://github.com/iporza/pthread-with-circular-b..uffer