

**รายงานการทดลองที่ 3SA03**

Parallel Programming With OpenMP

**จัดทำโดย**

นายปณิธาน ดวงขวัญ

รหัส 5735512036 Section 01

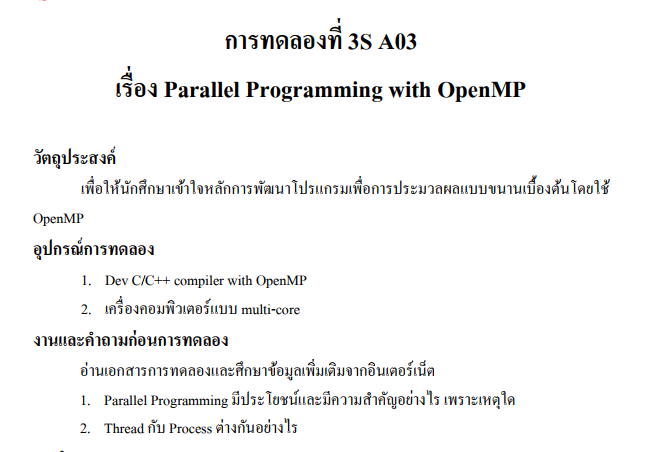
**เสนอ**

อาจารย์ฐิตินันท์ เกลี้ยงสุวรรณ

รหัสวิชา 242-301 Advanced Computer Engineering Laboratory I

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์



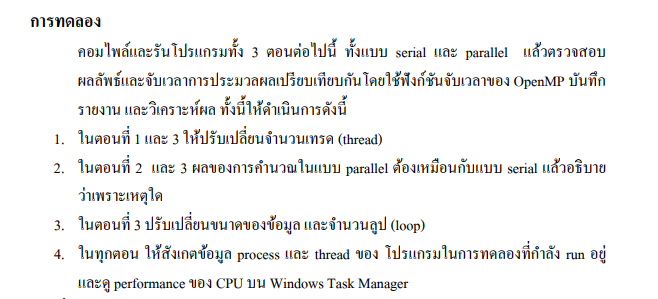
**คำถามก่อนการทดลอง**

1. **Parallel Programming มีประโยชน์และมีความสำคัญอย่างไร เพราะเหตุใด**

**ตอบ** โปรแกรมแบบขนานนั้นมีประโยชน์ในการทำงานในส่วนของโปรแกรมที่มีขนาดใหญ่และซับซ้อนเพราะการทำงานแบบขนานนั้นจะนำ thread หลายๆส่วนมาใช้งานเพื่อที่จะให้โปรแกรมทำงานได้เร็วและดีที่สุดโดย thread นั้นจะช่วยกันทำงานเพื่อให้โปรแกรมที่มีความซับซ้อนนั้นเสร็จได้อย่างมีประสิทธิภาพ

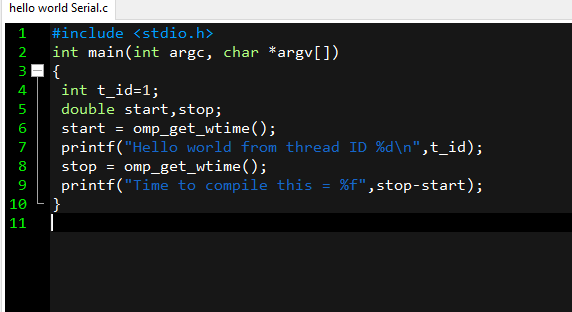
1. **Thread กับ Process ต่างกันอย่างไร**

**ตอบ** **Thread** คือ หน่วยทำงานของ Process ซึ่ง Thread นั้นจะต้องรันอยู่ภายใต้ process ส่วน **Process** นั้นคือ โปรแกรมที่กำลังถูกประมวลผล หรือ กระบวนการทำงานของโปรแกรมใดโปรแกรมหนึ่ง

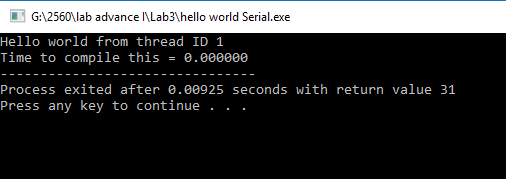


**ตอนที่ 1 Hello world**

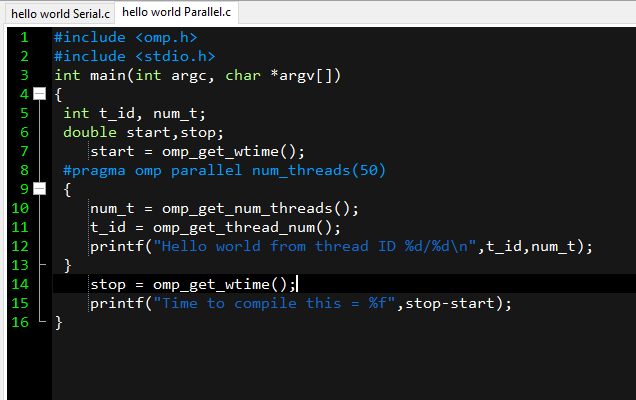
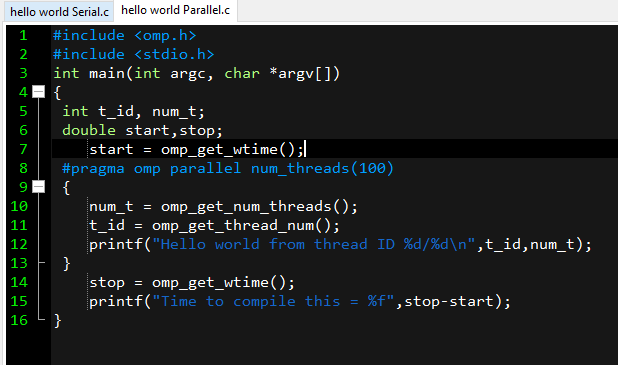
**Serial**



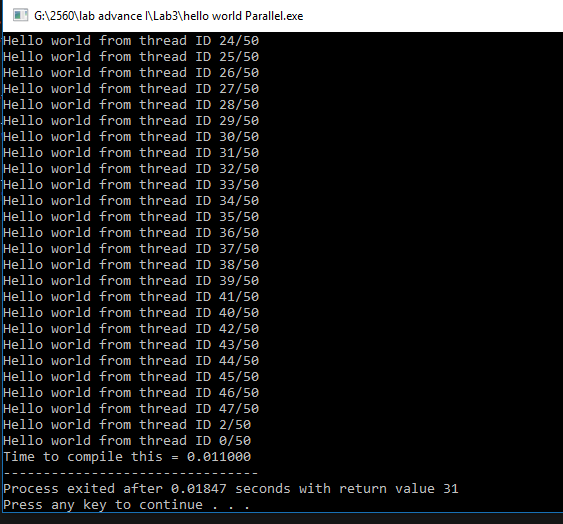
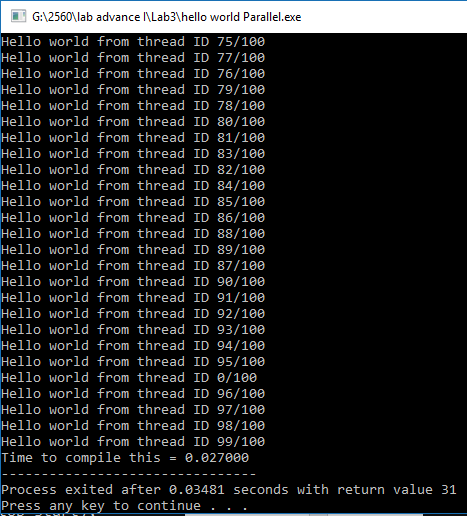
**ผลการ RUN**



**Parallel**

**ผลการ RUN**

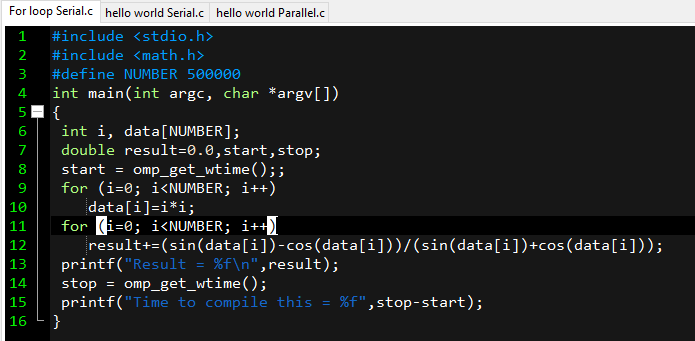
 

**สรุปผลการทดลอง**

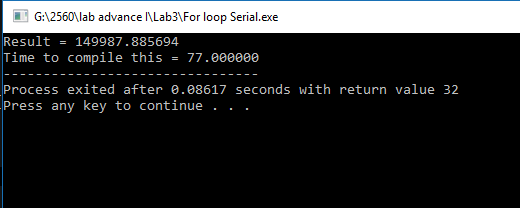
จากการทดลองจะพบว่าถ้าการทำงานแบบ Serial จะพบว่าคอมไพเลอร์จะใช้เวลาน้อยมากในการประมวลผลซึ่งต่างจากการทำงานแบบขนาน Parallel โดยการทำงานแบบขนาดสามารถใช้จำนวน thread ในการทำงานได้จากภาพประกอบด้านบนจะพบว่า ถ้าเรากำหนด thread น้อยจะทำให้เวลาในการประมวลน้อยลงเช่นกันโดยจะสรุปได้ว่า ถ้าเรากำหนดจำนวน thread เยอะเวลาที่ใช้ในการประมวลผลก็จะเยอะตามขึ้นมา โดยสามารถดูได้จากผลการรันและภาพประกอบด้านบน

**ตอนที่ 2 For loop**

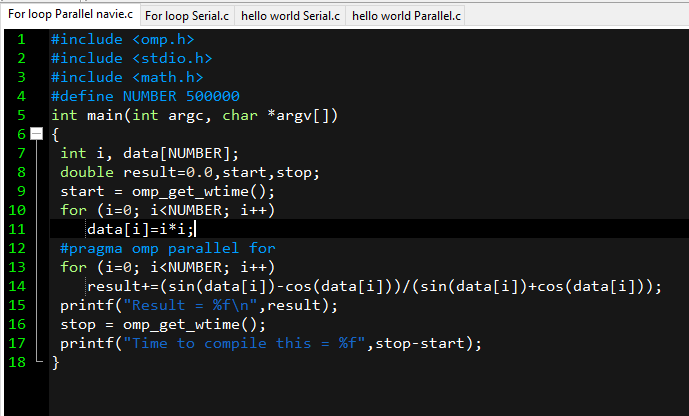
**Serial**



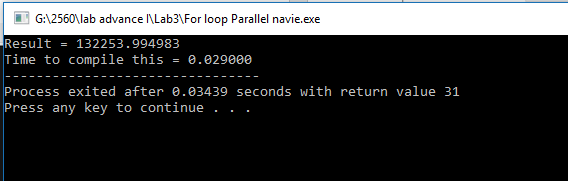
**ผลการ RUN**



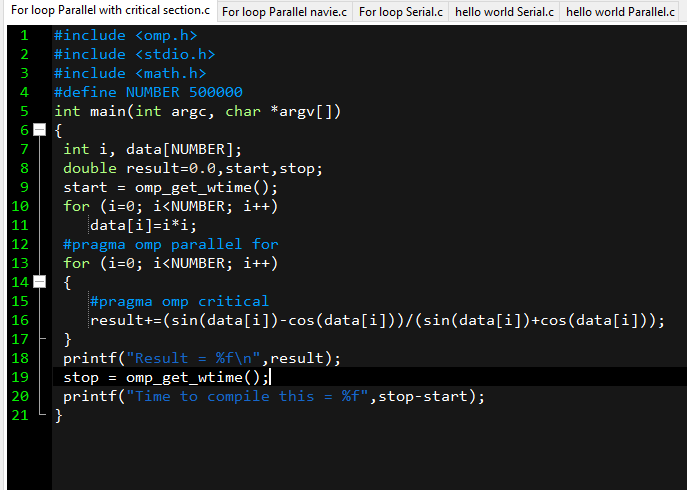
**Parallel ( naïve )**



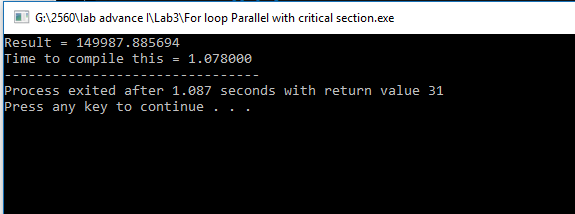
**ผลการ RUN**



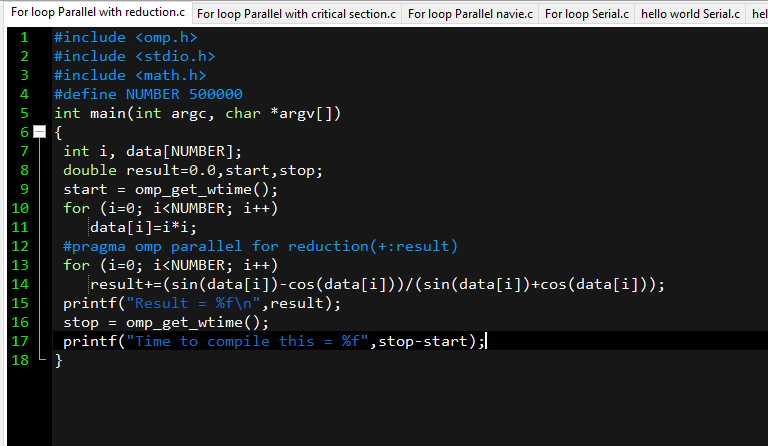
**Parallel with critical section**



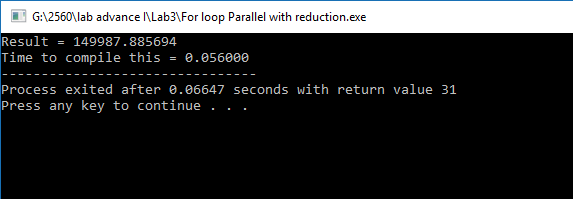
**ผลการ RUN**



**Parallel with reduction**



**ผลการ RUN**



**สรุปผลการทดลอง**

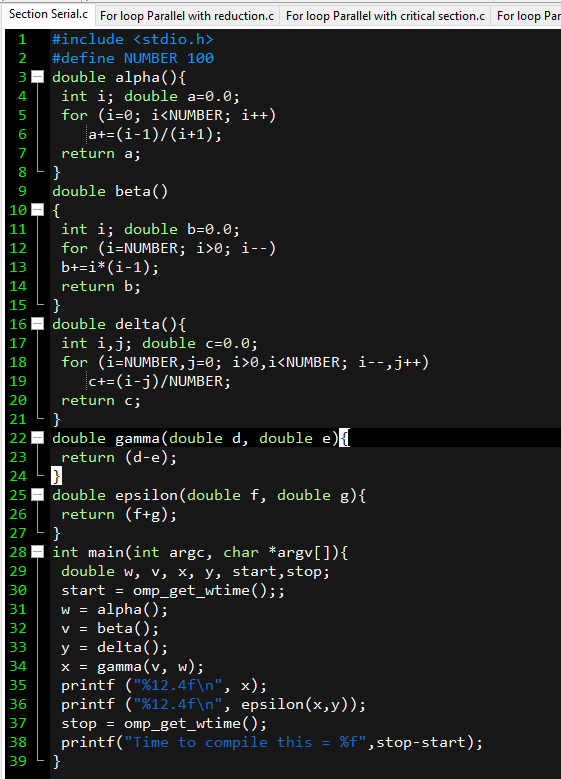
|  |  |  |
| --- | --- | --- |
| ชนิดการประมวลผล | Result | Time(s) |
| Serial | 149987.885694 | 0.056000 |
| Parallel (navie) | 86271.926770 | 0.038000 |
| Parallel with critical section | 149987.885694 | 0.887000 |
| Parallel with reduction | 149987.885694 | 0.041000 |

จากการทดลองจะพบว่าชนิดประมวลผลแต่ละชนิดจะได้ค่าที่ใกล้เคียงกันยกเว้น Parallel ( navie ) ที่จะพบว่าค่าทีได้ออกมานั้นแตกต่างจากการประมวลชนิดอื่นได้ชัดเจน ในส่วนของเวลาในการประมวลผลนั้นจะพบได้ว่าแบบ Parallel ( navie ) นั่นจะเร็วที่สุด ซึ่งที่เร็วรองลงมานั้น Parallel with reduction แต่ในส่วนของ Serial นั้นใช้เวลาในการประมวลผลนั้นปกติทั่วไป ส่วนในรูปแบบการประมวลผล critical section นั้นจะช้าที่สุดเพราะจะทำงานในส่วนของ Critical Section นั่นเองจึงทำให้ทำงานช้า

โปรแกรมทั้ง 3 รูปแบบนั้นประมวลผลได้ไม่ตรงกันโดยแบบ navie นั้นจะเป็นการทำงานโดยดูว่า thread ไหนว่างก็จะนำ thread นั้นไปทำงานหรือไปประมวลผล ส่วนแบบ critical จะเป็นการทยอยทำงาน ประมวลผล ซึ่งจะทำให้ใช้หลาย thread ในการทำงานซึ่งจะทำให้ประมวลผลช้านั่นเองและสุดท้ายแบบ reduction นั้นเป็นการให้ thread เข้าคิวประมวลผลทำงานซึงจะทำงานทีละ thread นั่นเอง

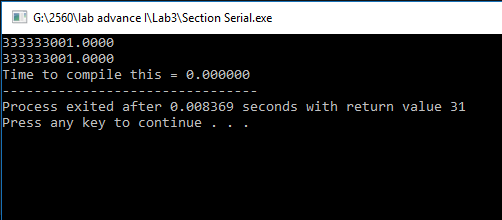
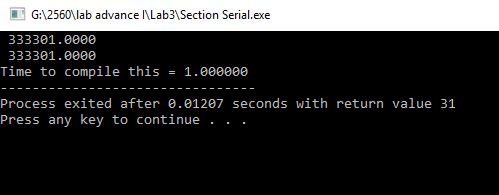
**ตอนที่ 3 Section**

**Serial**

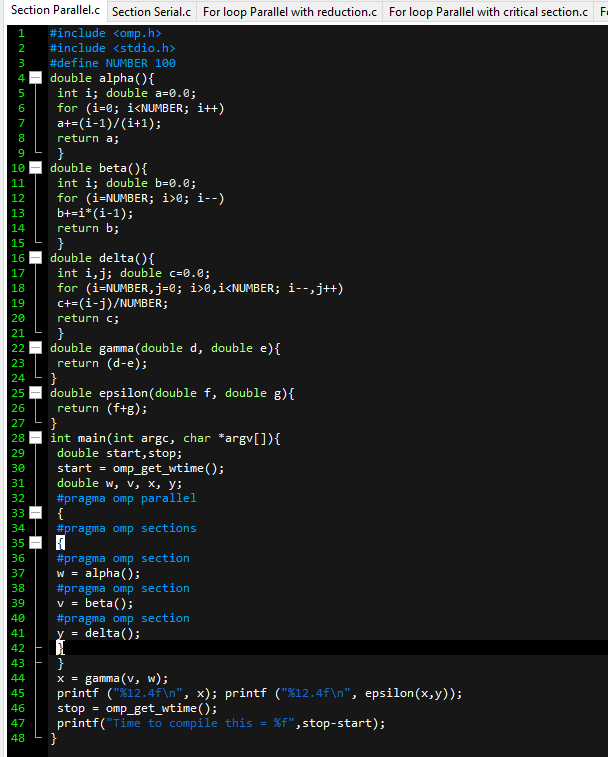


**ผลการ RUN**

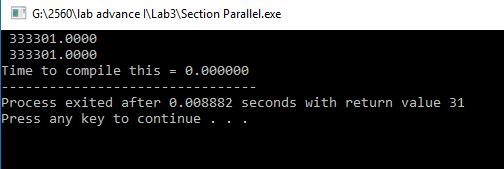
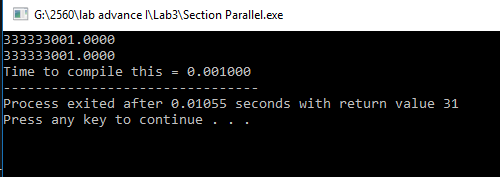
#define NUMBER 1000



**Parallel**



**ผลการ RUN**

#define NUMBER 1000

**สรุปผลการทดลอง**

|  |  |  |  |
| --- | --- | --- | --- |
| **ชนิดการประมวลผล** | **#define** | **Result** | **Time(s)** |
| **Serial** | **100** | **333301.0000** | **0.000000** |
| **Parallel** | **100** | **333301.0000** | **0.000000** |
| **Serial** | **1000** | **333333001.0000** | **0.001000** |
| **Parallel** | **1000** | **333333001.0000** | **0.001000** |

จากการทดลองจะพบว่า การปะมวลผลแบบ Serial ใน #define 1000 จะใช้เวลาการรันอยู่ที่ 0.001 และการประมวลผลแบบ Parallel ใน #define 1000 นั้นจะใช้เวลาเดียวกันซึ่งจริงๆในเวลาที่ทำที่แลปกับที่บ้านนั้นจะได้เวลาที่แตกต่างกันตามจริงแล้วเราจะพบว่าการประมวลผลแบบ Serial จะประมวลผลเร็วกว่า Parallel นั่นเอง

**คำถามท้ายการทดลอง**



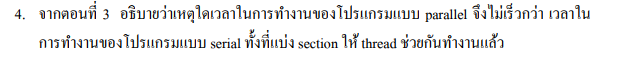
**คำตอบ** โปรแกรม Serial นั้นจะมีแค่ 1 thread ในการทำงานซึ่งจะเหมาะกับการทำงานที่ไม่ซับซ้อนหรือมีขั้นตอนการทำงานที่น้อยกว่า ส่วนโปรแกรมแบบขนาดนั้นจะมี thread มากกว่าหนึ่ง thread นั้นมันจะให้การทำงานนั้นเป็นขั้นเป็นตอนมากกว่า thread เดียวนั่นเองและเหมาะกับการทำงานของโปรแกรมที่มีขนาดใหญ่ หรือ ซับซ้อนเยอะๆนั่นเอง



**คำตอบ** ในการทำงานแบบขนาดโปรแกรมจะทำงานโดยไม่มีการเรียงลำดับในการงานจึงทำให้ขั้นตอนในการประมวลผลนั้นมีข้อผิดพลาดเกิดขึ้นเยอะจึงทำให้โปรแกรมทั้ง 3 รูปแบบนั้นประมวลผลได้ไม่ตรงกันโดยแบบ navie นั้นจะเป็นการทำงานโดยดูว่า thread ไหนว่างก็จะนำ thread นั้นไปทำงานหรือไปประมวลผล ส่วนแบบ critical จะเป็นการทยอยทำงาน ประมวลผล ซึ่งจะทำให้ใช้หลาย thread ในการทำงานซึ่งจะทำให้ประมวลผลช้านั่นเองและสุดท้ายแบบ reduction นั้นเป็นการให้ thread เข้าคิวประมวลผลทำงานซึงจะทำงานทีละ thread นั่นเอง



**คำตอบ** เนื่องจากโปรแกรมแบบขนานนั้นจะมีการแบ่ง Thread การทำงานเป็นหลายๆ thread โดยแต่ละ thread ก็จะไปประมวลผลโดยทำงานของตัวเอง อย่างเช่น thread นี้มีหน้าที่ในการทำงานตรงนี้ ก็จะทำแค่ตรงนี้นั่นเอง



**คำตอบ** การทำงานในแบบขนานนั้นจะใช้ thread หลาย thread มาช่วยในการทำงานซึ่งแตกต่างกับ Serial เพราะ Serial นั้นจะทำงานเพียง thread เดียวซึ่งทำงานเร็วกว่าหลาย thread แน่นอน