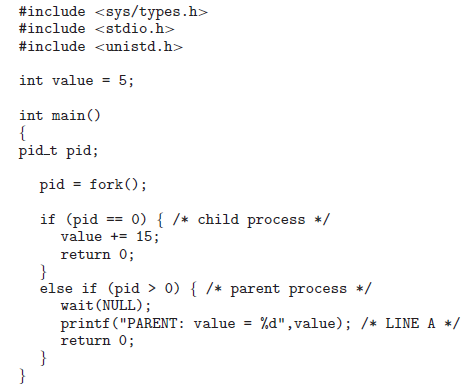
นาย อนุชิต ศุภวาลย์ 5907101003

**แบบฝึกหัด บทที่ 3**

**1. (10 คะแนน) จากโค้ดภาษาซีด้านล่างต่อไปนี้ โปรแกรมที่บรรทัด Line A ให้เอาท์พุตเป็นค่าใด จงทดลองและอธิบายผลการทดลอง**



* ฟังก์ชันfork( ) ได้สร้างโปรเซสขึ้นมาโดย ฟังก์ชันแรก

if (pid = = 0) {

value += 15;

return 0;

}

ถ้า pid = 0 ฟังก์ชันจะทำงาน ทำการเพิ่มค่า 15 ในกับ vealue แล้ว return 0

else if (pid > 0) {

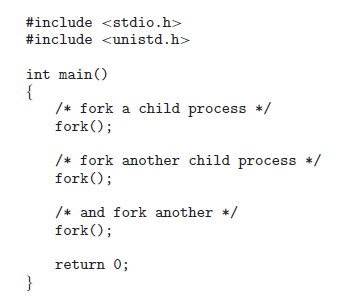
wait (NULL);

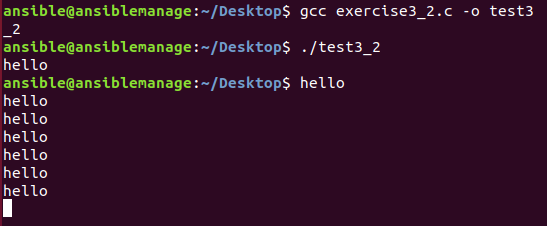
printf("Value = %d", value);

return 0;

ถ้า pid > 0 ค่าเดิมขอ value = 5 และรันค่าทั้งหมด จะได้ผลรับคือ 0

**2. (5 คะแนน) จากโค้ดภาษาซีต่อไปนี้ มีโปรเซสทั้งหมดกี่โปรเซส (นับรวมโปรเซสเริ่มต้นด้วย) จงอธิบาย**





* มีโปรเซสทั้งหมด 8 โปรเซส โดยมีค่าเท่ากับ 2 โดยในที่นี้มีการทำงานโดย fork( ) มีการทำงานถึง 3 ครั้ง จึงได้ 2\*2\*2 หรือ 2^3 = 8

**3. (5 คะแนน) เมื่อมีการสร้างโปรเซสใหม่โดยใช้คำสั่ง fork มีการแชร์สถานะ ( State ) ของสิ่งใดต่อไปนี้ Heap , Stack , และ Shared Memory Segment ระหว่างโปรเซสแม่กับโปรเซสลูก**

**-** มีการแชร์สถานะ shared memory segment / Shared memory เป็นพื้นที่หน่วยความจำซึ่งโปรเซสต่างๆสามารถใช้งานร่วมกันได้ โดยโปรเซสต่างๆจะใช้ shared memory ในการติดต่อสื่อสารกัน

**4. (5 คะแนน) จงอธิบายความหมายของ Short-Term , Medium-Term และ Long-Term Scheduling**

**- Short-term**

* + ทำหน้าที่คัดเลือกกระบวนการที่พร้อมจะถูกกระทำการขึ้นมาหนึ่งกระบวนการให้เข้าใช้ซีพียู
* **Long-term** 
  + ทำหน้าที่คัดเลือกกลุ่มกระบวนการขึ้นมาจากที่พักเก็บไปยังหน่วยความจำหลักไปยังคิวพร้อม
  + ควบคุมดีกรีของการทำงานแบบหลายโปรแกรม (degree of multiprogramming) ให้เหมาะสม
* **medium-term scheduling**
  + ทำหน้าที่ขจัดกระบวนการออกจากหน่วยความจำ ช่วยลดดีกรีของการทำงานแบบหลายโปรแกรมลง

**5. (5 คะแนน) จงอธิบายสิ่งที่ Kernel ทำเมื่อเกิด Context - Switch ระหว่างโปรเซส**

- Kernel คือแก่นของระบบปฏิบัติการเป็นส่วนหนึ่งของระบบปฏิบัติการที่จะถูกโหลดขึ้นมาเป็นอันดับแรกและถูกเก็บไว้ ในหน่วยความจำหลักมีหน้าที่จัดสรรทรัพยากรให้ใช้ทรัพยากรต่าง ๆ รวมถึงอุปกรณ์ควบคุมฮาร์ดแวร์

- Context Switching คือ การที่ระบบปฏิบัติการเปลี่ยนการประมวลผลจากโปรเซสหนึ่งไปประมวลผลอีกโปรเซส

- คือ ระบบปฏิบัติการจะหยุดประมวลผลโดยโปรเซสหนึ่ง ซึ่งปัจจุบันมีสถานะของโปรเซสเป็น Running

และไปประมวลผลอีกโปรเซสหนึ่งซึ่งอยู่ในสถานะ Ready แทน

**6. (30 คะแนน) โปรเซส init ในระบบปฏิบัติการ Unix หรือ Linux คืออะไร มีหน้าที่อย่างไรจงอธิบายพอสังเขป**

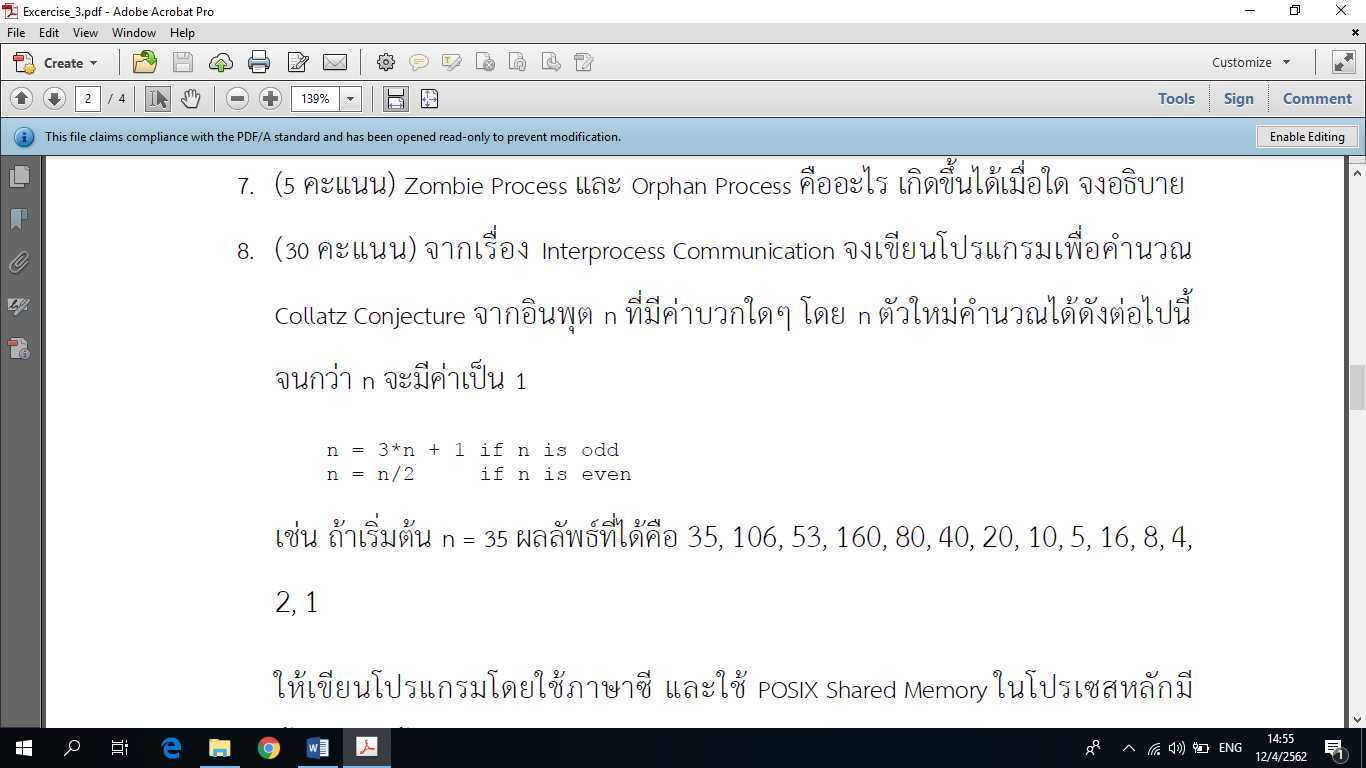
- [Init](http://www.tutorialspoint.com/#3)เป็นพาเรนต์ของกระบวนการทั้งหมด บทบาทหลักคือการสร้างกระบวนการจากสคริปต์ที่เก็บไว้ในไฟล์ / etc / inittab แฟ้มนี้มักจะมีรายการที่ทำให้เกิด init จะวางไข่ Getty s ในแต่ละบรรทัดที่ผู้ใช้สามารถเข้าสู่ระบบ. นอกจากนี้ยังควบคุมกระบวนการอิสระที่จำเป็นโดยระบบใด ๆ

**7. (5 คะแนน)** **Zombie Process และ Orphan Process คืออะไร เกิดขึ้นเมื่อใด จงอธิบาย**

- zombie คือ  กระบวนการ zombie คือโปรเซสตายที่ไม่ได้ดำเนินการอีกต่อไป แต่ยังคงเป็นที่รู้จักในตารางโปรเซส ไม่มีการจัดสรรพื้นที่ว่างระบบอื่นให้กับโปรเซสชนิดนี้ กระบวนการ zombie ถูกทำลาย หรือออกและมีอยู่ต่อไปในตารางโปรเซสจนกว่า Parent Process ตายหรือปิดและรีสตาร์ทระบบ กระบวนการ zombie แสดงเป็น <defunct> เมื่อแสดงรายการโดยคำสั่ง ps สาเหตุก็คือ Parent Process ยังไม่ได้รับ system call จาก Child Process จึงทำให้เกิด Zombies Process ขึ้นมา แถม Process นั้นยังกิน Memmory ของเครื่องอีกต่างหาก

- orphan เป็นกระบวนการคอมพิวเตอร์ที่มีกระบวนการหลักได้เสร็จสิ้นหรือสิ้นสุดแม้ว่ามันจะยังคงทำงานอยู่ เคสที่ parent ทำการ terminateไปแล้วในขณะที่ child ยัง execute ไม่เสร็จและไม่เกิดการ cascading (os อนุญาติให้เด็กมีชีวิตต่อไป) กรณีที่เราเรียก child ว่า orphan

**8. (30 คะแนน) จากเรื่อง Interprocess Communication จงเขียนโปรแกรมเพื่อคำนวณ Collatz Conjecture จากอินพุต n ที่มีค่าบวกใด ๆ โดย n ตัวใหม่คำนวณได้ดังต่อไปนี้ จนกว่า n จะมีค่าเป็น 1**

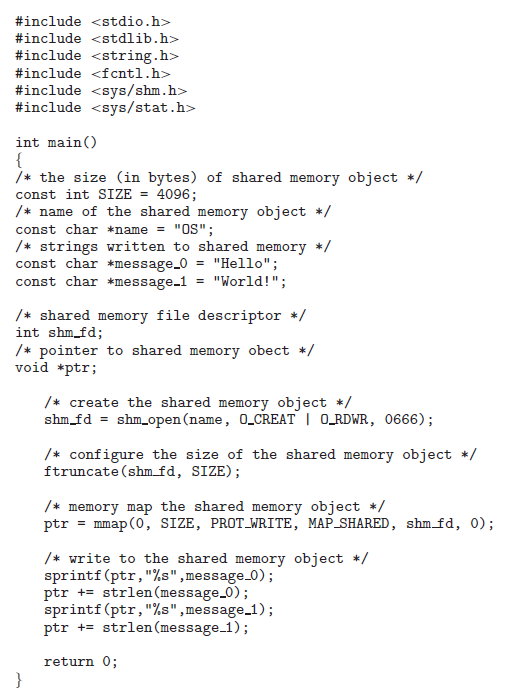
****

**เช่น ถ้าเริ่ม n = 35 ผลลัพธ์ที่ได้คือ 35 , 106 , 53 , 160 , 80 , 40 , 20 , 10 , 5 , 16 , 8 , 4 , 2 , 1**

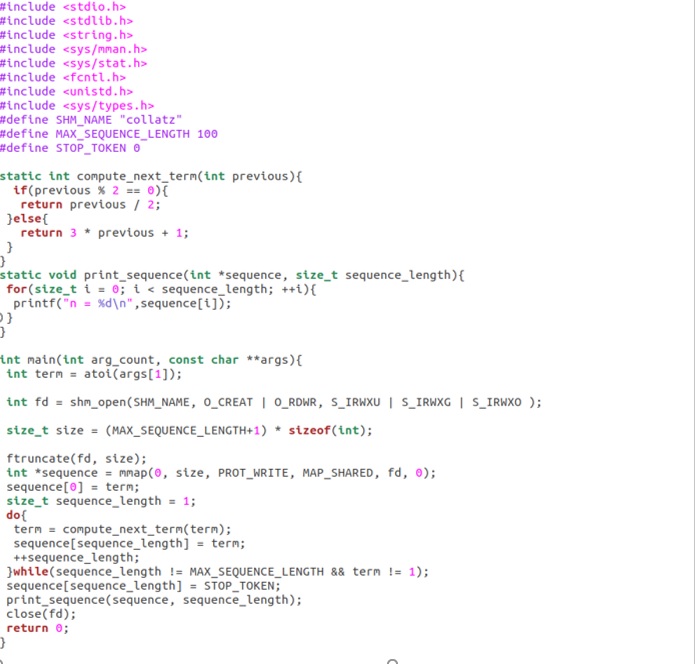
**ให้เขียนโปรแกรมโดยใช้ภาษาซี และใช้ POSIX Shared Memory ในโปรเซสหลักมีขั้นตอนดังนี้**

1. **สร้าง Shared-Memory Object ( shm open () , ftruncate () , and mmap () )**
2. **สร้างโปรเซสลูกเพื่อคำนวณ n โดยโปรเซสลูกเขียนค่า n ที่คำนวณได้ในแต่ละรอบลงใน Shared-Memory Object**
3. **รอจนโปรเซสลูกเสร็จ แล้วแสดงผลขอมูลจาก Shared-Memory Object ออกทางจอภาพ**
4. **ลบ Shared-Memory Object**

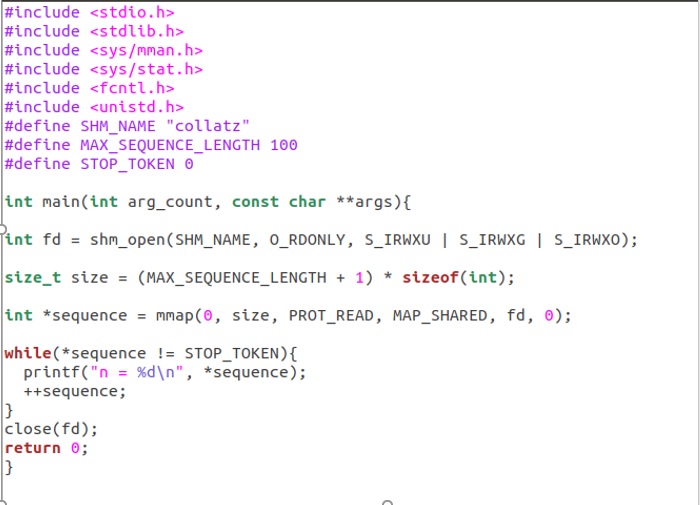
**ตัวอย่างการใช้ Shared-Memory Object**



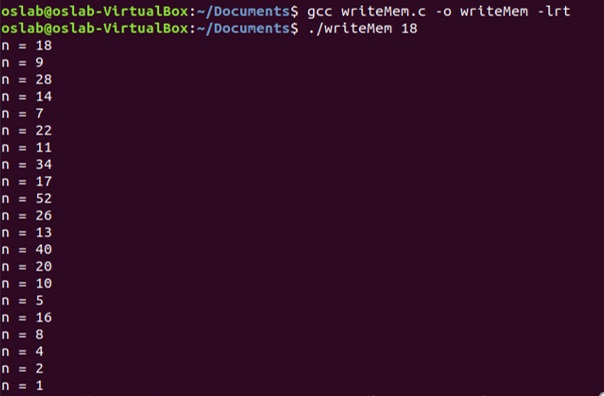
writeMem.c



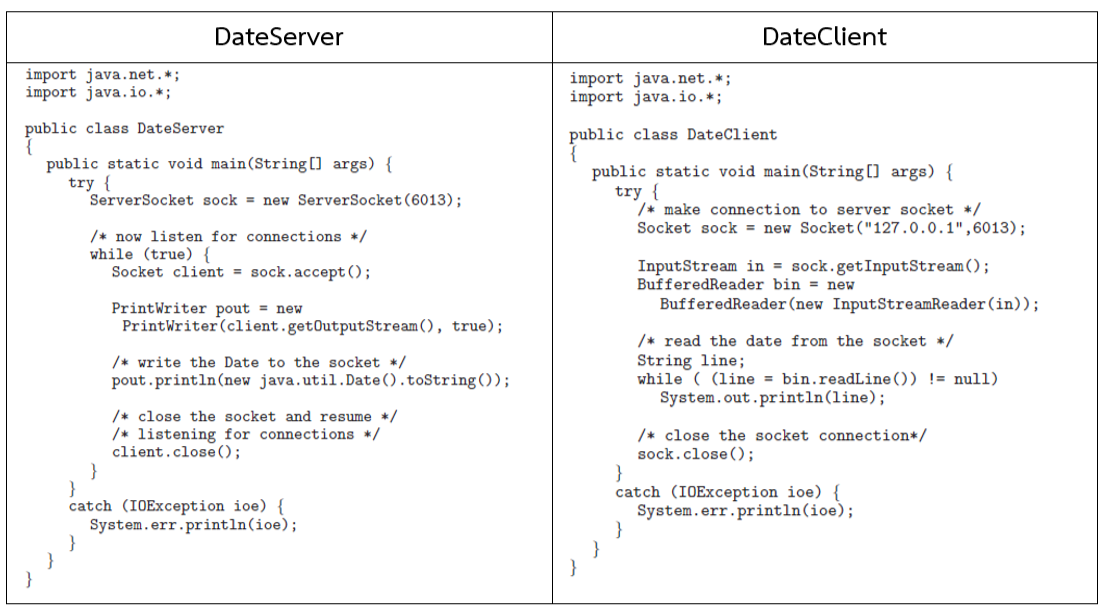
readMem.c



ผลลัพธ์



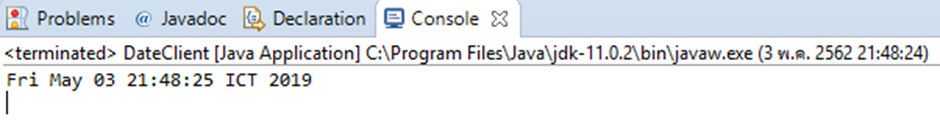
**9.** **(15 คะแนน)** **จากเรื่อง Interprocess Communication จงทดลองโปรแกรมภาษาจาวาต่อไปนี้ ( หมายเหตุ ให้รัน DateSever ก่อน แล้วจึงค่อยรัน DateClient )**



**a. โค้ดตัวอย่างข้างต้นมี Interprocess Communication ด้วยวิธีการใด**

- ด้วยวิธีการ socket

**b. การรันโปรแกรมได้ผลลัพธ์อะไร จงอธิบายพฤติกรรมของโปนแกรมอย่างละเอียด**

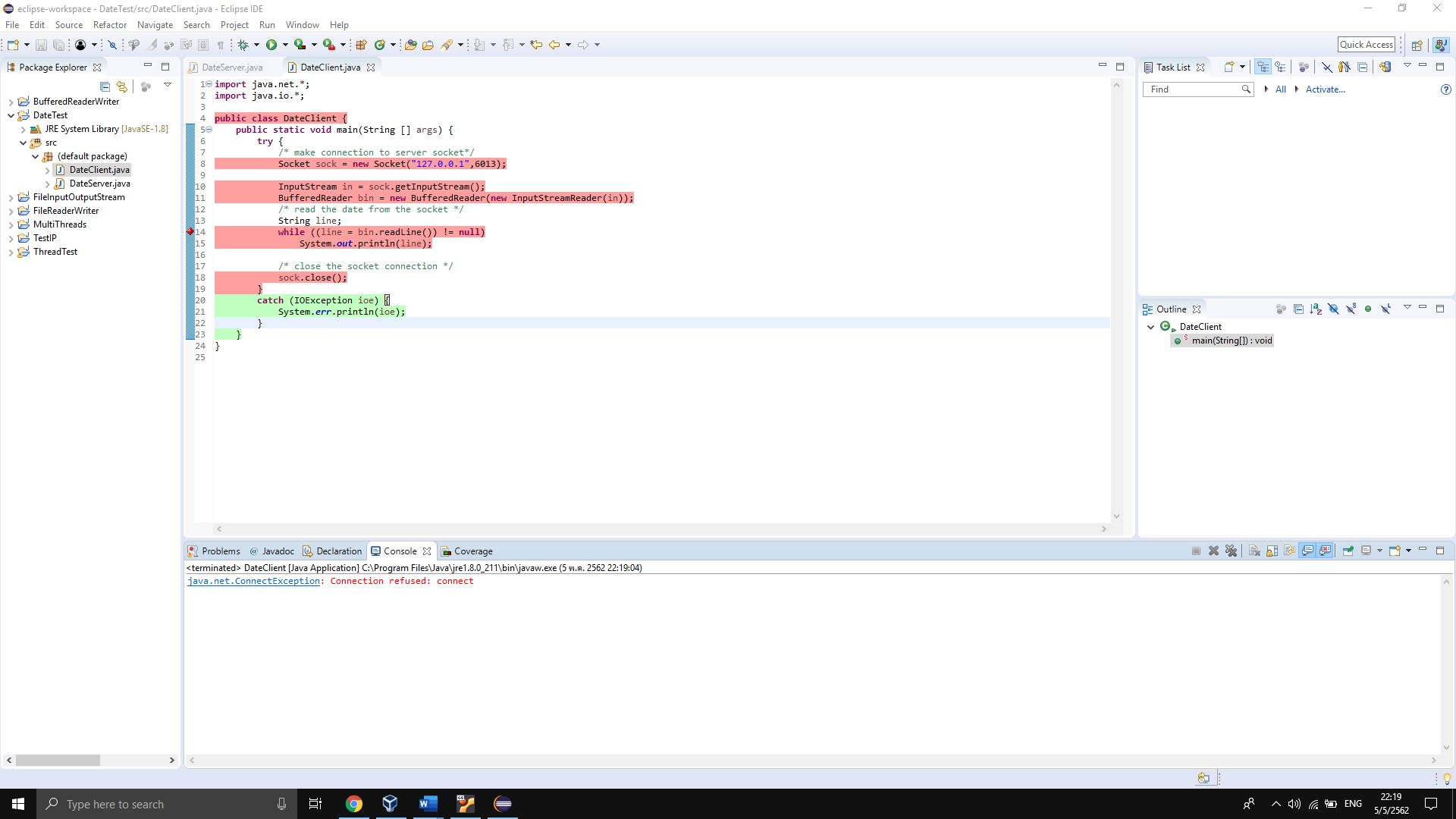


- DateSever จะเปิด port 6013 ไว้สำหรับการเชื่อมต่อ และ Date Client จะทำการเชื่อมต่อ

**c. จงอธิบายความหมายของ Socket**

- การจัดการการสื่อสารระหว่างกระบวนการคือการใช้ซ็อกเก็ต พวกเขาให้การสื่อสารแบบจุดต่อจุดสองทางระหว่างสองกระบวนการ ซ็อกเก็ตเป็นจุดสิ้นสุดของการสื่อสารและชื่อสามารถผูกไว้กับพวกเขา ซ็อกเก็ตสามารถเชื่อมโยงกับกระบวนการอย่างน้อยหนึ่งกระบวนการ

**d. หากรัน DateClient ก่อนการรัน DateServer ได้ผลลัพธ์อย่างไร**



DateClient ไม่สามารถเชื่อมต่อกับ DateServer เนื่องจาก DateServer ยังไม่ได้ทำการเปิดพอทเชื่อมต่อไว้