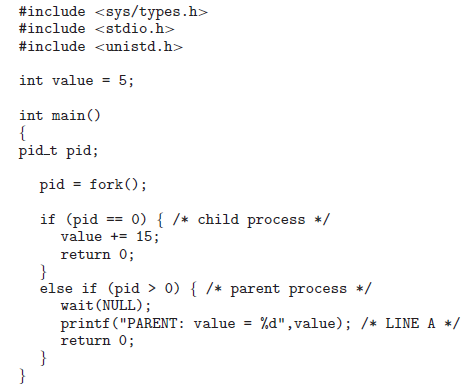
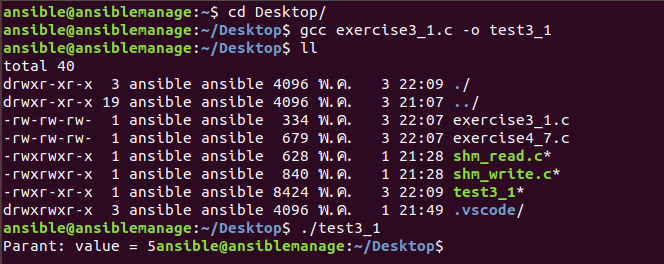
5907101011 นายสิรวิชญ์ เตชะวณิชย์

**Exercise 3**

1. **จากโค้ดภาษาซีด้านล่างต่อไปนี้ โปรแกรมที่บรรทัด Line A ให้เอาท์พุตเป็นค่าใด จงทดลองและอธิบายผลการทดลอง**





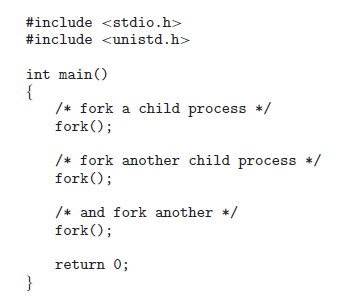
*( ภาพประกอบจากผลการทดลอง )*

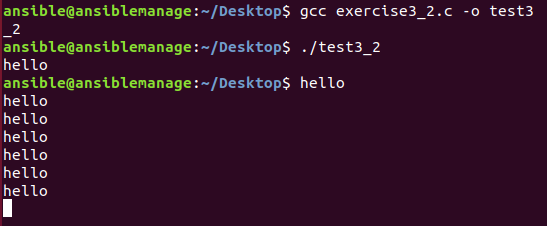
หาก pid มีค่าเท่ากับ 0 จะทำที่ child process ถ้ามากกว่า 0 จะทำที่ child process ก่อนแล้วจึงทำ parent process

ที่บรรทัด LINE A Parent process มีการ print แสดงผลค่าของ Parent ออกมามีค่าเท่ากับ 5

เนื่องจาก value ที่เข้ามามีการตั้งค่าไว้ที่ 5 ส่วนของ child นั้นจะมีค่า 20 เนื่องจากการ fork เพียง 1 ครั้งแต่ในส่วนของ child process ไม่ได้มีการใช้คำสั่ง print ออกมากเพือแสดงผล

1. **จากโค้ดภาษาซีต่อไปนี้ มีโปรเซสทั้งหมดกี่โปรเซส (นับรวมโปรเซสเริ่มต้นด้วย) จงอธิบาย**





*( ภาพประกอบจากผลการทดลอง )*

จำนวนของ process ทั้งหมดสามารถหาได้จากจำนวนการ fork หากจำนวน fork = n จะมีจำนวน process เท่ากับ 2n ในกรณีตัวอย่างนั้นมีการ fork 3 ครั้งจะทำให้เกิด process ทั้งหมดคือ 8 process ( 2\*2\*2 = 8)

f1 /\* fork child process (f1) fork (); f1\*/

/ \

f2 f2 /\* fork child process (f2) fork (); f2\*/

/ \ / \

f3 f3 f3 f3 /\* fork child process (f3) fork (); f3 \*/

/ \ / \ / \ / \

1 2 3 4 5 6 7 8 /\* output \*/

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

มีการแชร์เฉพาะส่วนของ Memory Segment ระหว่าง parent process กับ child parent

1. **จงอธิบายความหมายของ Short-Term , Medium-Term และ Long-Term Scheduling**

Short-term Scheduling คือ ตัวจัดตารางระยะสั้นหรือตัวกำหนดเวลา CPU มีหน้าที่ในการเลือก Process ที่อยู่ในสถานะรอเพื่อเข้าใช้หน่วยประมวลผลกลางประมวลผลต่อไป มักถูกเรียกใช้อยู่บ่อยครั้งและเป็นแค่เพียงช่วงเวลาสั้นๆเท่านั้น

Medium-term Scheduling คือ ตัวจัดการตารางระยะกลาง คือ ย้าย Process ออกจากหน่วยความจำหลักเพื่อลดจำนวน Process ที่มีมากเกินไปในหน่วยความจำ เพื่อทำให้ CPU ทำงานได้ดีและเร็วขึ้นช่วยในส่วนของการ balance ระหว่าง processมีการสลับการทำงานไม่มากเท่า short-term

Long-term Scheduling คือ ตัวจัดตารางระยะยาว คือ เลือก Process จากหน่วยเก็บข้อมูลเพื่อเข้าสู่หน่วยความจำ การจัดตารางระยะยาวเกี่ยวข้องกับการเลือกกระบวนการจากหน่วยเก็บข้อมูลในหน่วยความจำสำรอง และ จัดลงใน คิวที่อยู่ในสถานะพร้อม ในหน่วยความจำหลัก เพื่อ รอดำเนินการ ที่จะถูกนำไปประมวลผลต่อไป

1. **จงอธิบายสิ่งที่ Kernel ทำเมื่อเกิด Context - Switch ระหว่างโปรเซส**

เมื่อเกิด Context - Switch จะทำการบันทึก save Process ที่ประมวลผลปัจจุบันที่อยู่สถานะ Runnung เอาไว้ และต่อมาจะทำการเรียก load Process อื่นๆหรือ Process ที่ถูกบันทึกอยู่สถานะ Ready ไว้กลับขึ้นมาทำประมวลผลต่ออีกครั้งและเมื่อ ประมวลผล Process ที่ทำอยู่เสร็จก็จะสลับไปกลับยัง Process อื่นๆที่อยู่ถัดไปในคิว

1. **โปรเซส init ในระบบปฏิบัติการ Unix หรือ Linx คืออะไร มีหน้าที่อย่างไรจงอธิบายพอสังเขป**

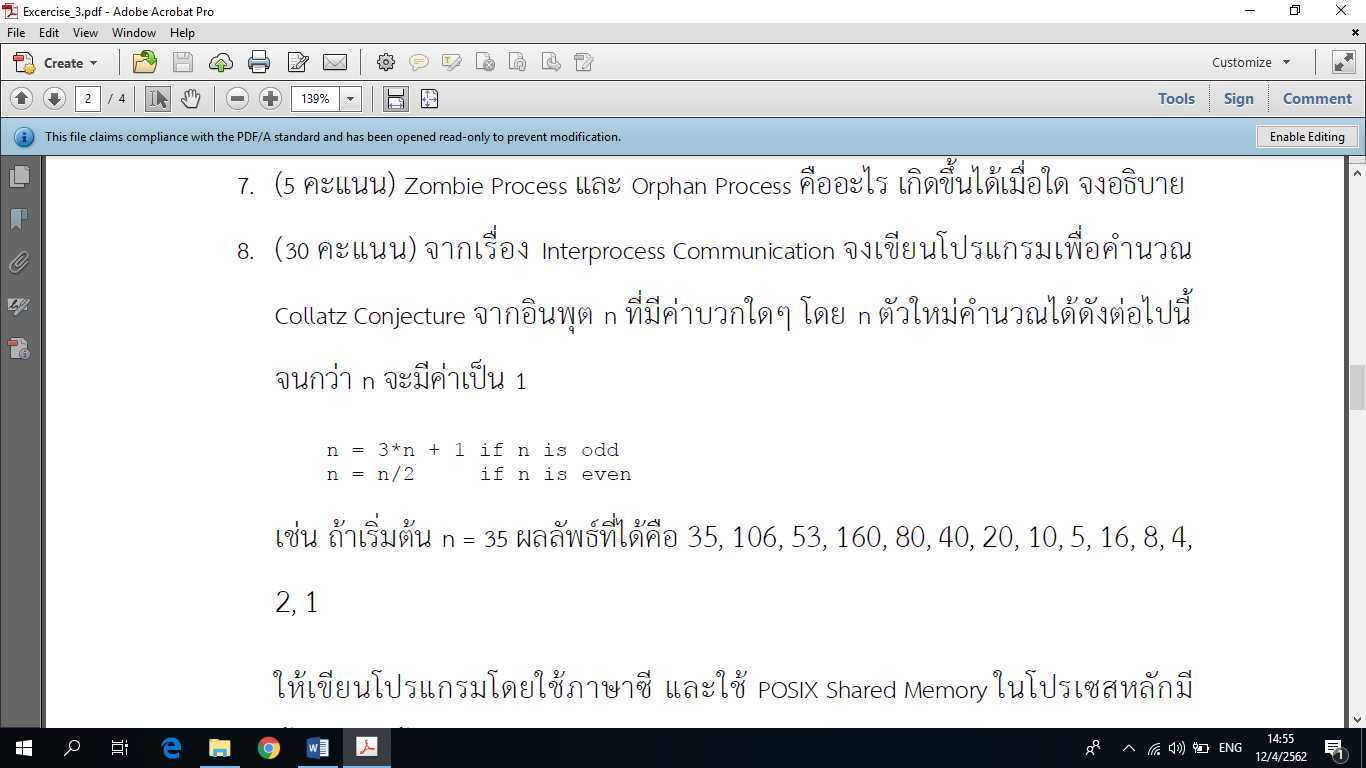
init (ย่อมาจาก initialization) เป็นกระบวนการที่ต่อเนื่องจาก Kernel boot ระหว่างการบูทระบบคอมพิวเตอร์ Init เป็น daemon process ที่ทำงานตั้งแต่บูทระบบเสร็จต่อไปจนกว่าระบบจะปิด มักรู้จักในชื่อ user space init process ถูกจัดอยู่ในกลุ่มของ operating system service-management

1. **Zombie Process และ Orphan Process คืออะไร เกิดขึ้นเมื่อใด จงอธิบาย**

Zombie process  หรือ process ผีดิบ คือ process ตายที่ไม่ได้ดำเนินการอีกต่อไป แต่ยังคงมี Process ID อยู่แม้จะมีการทำการคืนทรัพยากรและหน่วยความจำกลับคืนให้แก่ระบบแล้วแต่ก็ยังไม่ได้มีการจัดสรรพื้นที่ว่างแก่นั้น Process อื่นๆทำให้ทรัพยากรส่วนนั้นไม่สามารถถูกนำไปใช้ได้ แต่ไม่สามารถกำจัด Process นี้ออกจากระบบได้ เนื่องจาก Child process หรือ Zombie process นี้เข้าสู่สถานะ terminate โดยขณะนั้น Parent process ยังไม่ถึงในสถานะ waiting ทำให้ Parent process เมื่อเข้าสู่สถานะ waiting แล้วต้อง waiting ตลอดไป ไม่สามารถจัดการกับ process นี้ได้ ทำให้ไม่สามารถทำลาย process นี้ออกจากระบบ

Orphan Process หรือ process กําพร้า คือ process ที่ child process อยู่ในระหว่างการ execute ยังไม่สามารถเข้าสู่สถานะ terminate แต่ในส่วนของ Parent process กลับเข้าสู่สถานะ terminate ก่อนแม้จะมีความผิดพลาดเกิดขึ้นแต่ process นี้ยังคงสมารถกำจัดอยู่ในระบบต่อไปและสามารถกำจัดได้

1. **จากเรื่อง Interprocess Communication จงเขียนโปรแกรมเพื่อคำนวณ Collatz Conjecture จากอินพุต n ที่มีค่าบวกใด ๆ โดย n ตัวใหม่คำนวณให้ได้ดังต่อไปนี้ จนกว่า n จะมีค่าเป็น 1**

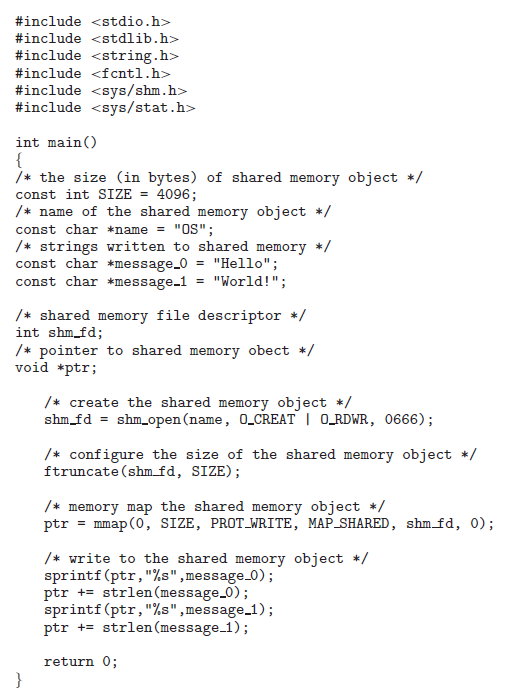
****

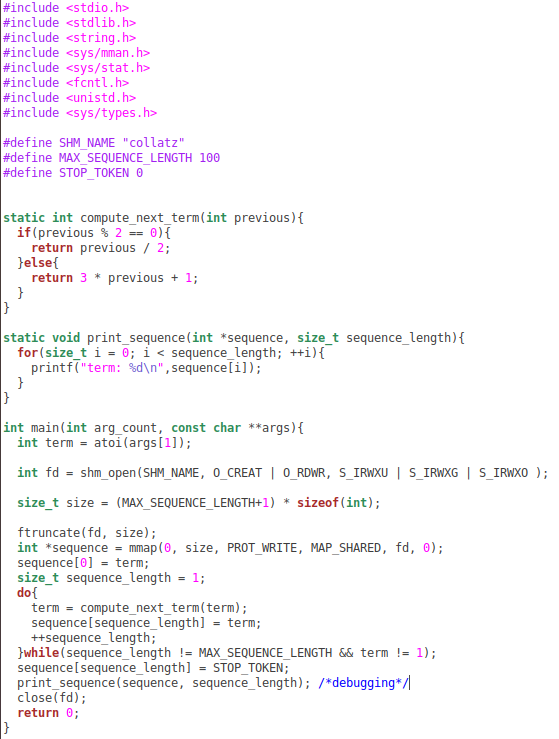
**เช่น ถ้าเริ่ม n = 35 ผลลัพธ์ที่ได้คือ 35 , 106 , 53 , 160 , 80 , 40 , 20 , 10 , 5 , 16 , 8 , 4 , 2 , 1**

**ให้เขียนโปรแกรมโดยใช้ภาษาซี และใช้ POSIX Shared Memory ในโปรเซสหลักมีขั้นตอนดังนี้**

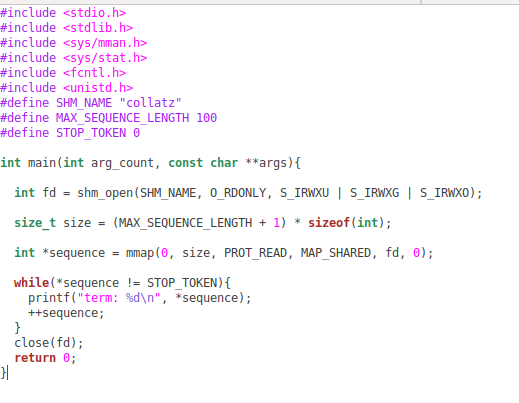
1. **สร้าง Shared-Memory Object ( shm open () , ftruncate () , and mmap () )**
2. **สร้างโปรเซสลูกเพื่อคำนวณ n โดยโปรเซสลูกเขียนค่า n ที่คำนวณได้ในแต่ละรอบลงใน Shared-Memory Object**
3. **รอจนโปรเซสลูกเสร็จ แล้วแสดงผลขอมูลจาก Shared-Memory Object ออกทางจอภาพ**
4. **ลบ Shared-Memory Object**

**ตัวอย่างการใช้ Shared-Memory Object**

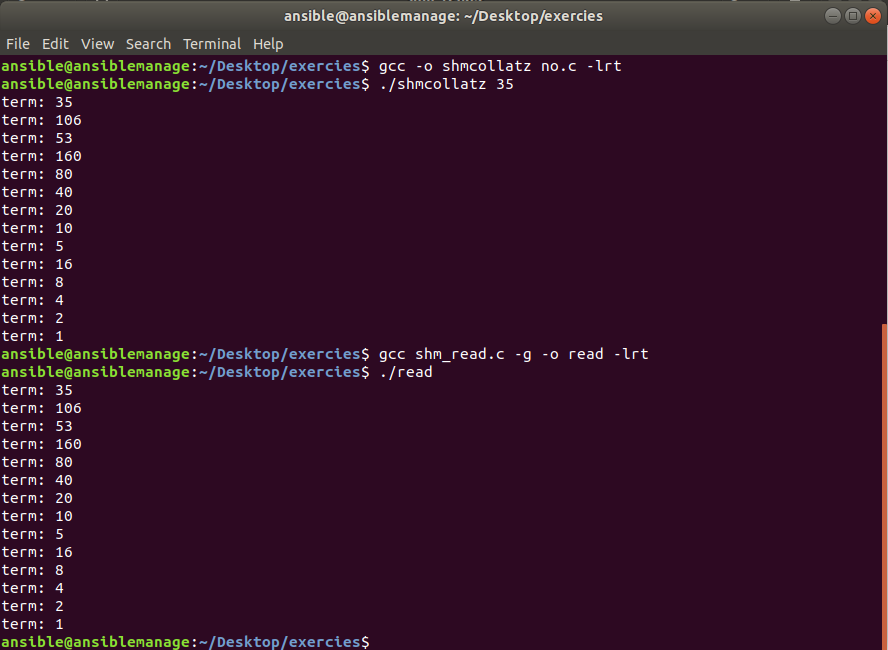




*( Sample code write to memory )*



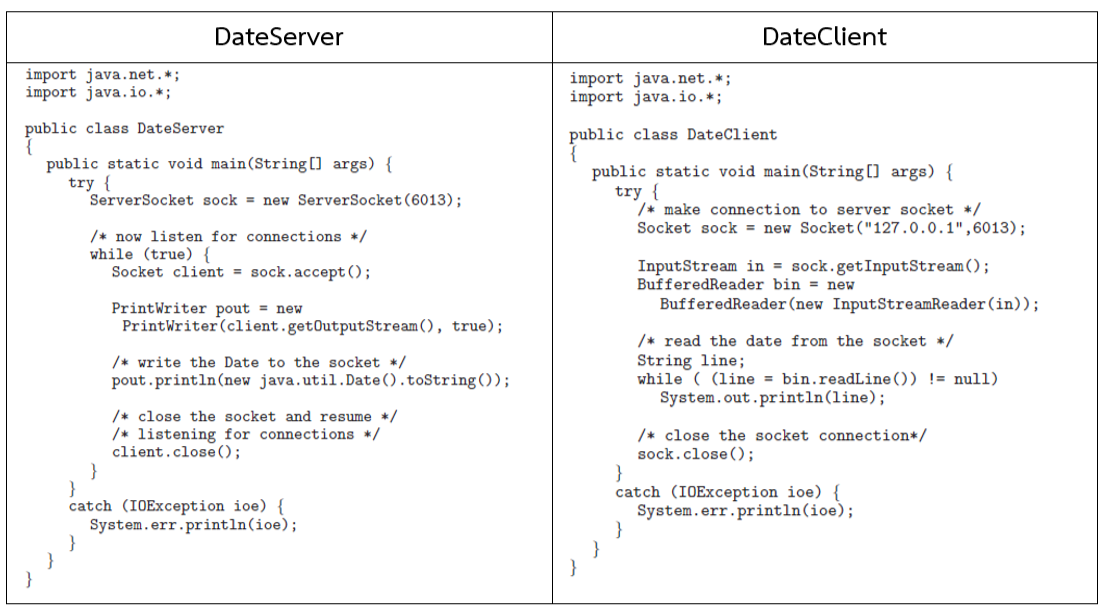
*( Sample code read from memory )*



*( ภาพประกอบจากผลการทดลอง )*

1. **จากเรื่อง Interprocess Communication จงทดลองโปรแกรมภาษาจาวาต่อไปนี้**

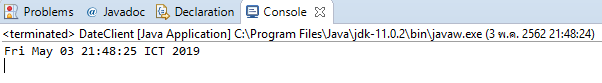
**( หมายเหตุ ให้รัน DateSever ก่อน แล้วจึงค่อยรัน DateClient )**



1. **โค้ดตัวอย่างข้างต้นมี Interprocess Communication ด้วยวิธีการใด**
2. **การรันโปรแกรมได้ผลลัพธ์อะไร จงอธิบายพฤติกรรมของโปนแกรมอย่างละเอียด**
3. **จงอธิบายความหมายของ Socket**
4. **หากรัน DateClient ก่อนการรัน DateServer ได้ผลลัพธ์อย่างไร**
5. **โค้ดตัวอย่างข้างต้นมี Interprocess Communication ด้วยวิธีการใด**

Code ข้างต้นมี Interprocess Communication ด้วยวิธีการแบบ Socket

1. **การรันโปรแกรมได้ผลลัพธ์อะไร จงอธิบายพฤติกรรมของโปนแกรมอย่างละเอียด**



พฤติกรรมของโปรแกรม DateServer จะทำการเปิด socket หรือ port 6013 ไว้เพื่อรอการเชื่อมจากภายนอกและรอที่จะส่งข้อมูลวันที่ในรูปแบบของ String ออกไปยัง Client ที่เชื่อมต่อ DateClient นั้น

จำลองเป็นผู้เชื่อมต่อซึ่งมาเชื่อมต่อใน socket หรือ port เดียวที่ตรงกับของ DateServer จึงได้รับข้อมูลวันที่จาก DateServer มาและได้ทำการปริ้นแสดงผลออกมา

ที่ class DateServer

* มีการสร้าง socket ขึ้นมาโดนระบุ port ที่ใช้คือ 6013
* มี Loop while ตรวจสอบว่ามีการเชื่อมต่อที่ socket นี้หรือไม่หากเป็นจริงก็จะทำงานต่อไป
* มีการสร้าง Object PrintWriter pout เพื่อทำการ get ค่า OutputStream จาก client ที่เชื่อมต่อ
* มีการใช้ Object pout เพื่อ เขียนข้อมูลวันที่ในรูปแบบของ String
* หลังจากนั้นทำการตัดการเชื่อมต่อกับ client ที่เข้ามาเชื่อต่อ port 6013 และทำการรอคอยการเชื่อมต่อใหม่อีกครั้ง
* มีการใช้ try catch เพื่อดักจับ ERROR

ที่ class DateClient

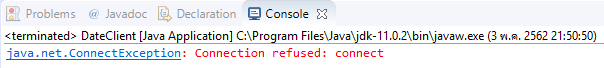
* มีการสร้าง Socket ขึ้นมา IP 127.0.0.1 Port 6013
* มีการสร้าง Object InputStream in เป็น get ค่า InputStream
* มีการสร้าง Object BufferedReader มี parameter เป็น Object InputStream(in)
* มีการสร้างตัวแปร Line ชนิด String
* มี Loop เพื่อเช็คว่าถ้า มีการเขียนค่า String ที่มาจาก BufferedReader ลงในตัวแปร Line ให้ทำการปริ้นข้อความนั้นออกมาแสดงผล
* หลังจากนั้นทำการตัดการเชื่อมต่อกับ socket นั้น
* มีการใช้ try catch เพื่อดักจับ ERROR

1. **จงอธิบายความหมายของ Socket**

ความหมายของ Socket คือ เป็น Interprocess Communication รูปแบบหนึ่งเพราะเป็นจุดเชื่อมต่อระหว่าง Process ที่ทำงานบนระบบเครือข่าย โดยProcess ใช้ Socket เป็นจุดรับส่งข้อมูลซึ่งจะทำการเปิด Port หรือ Socket หมายเลขใดๆหนึ่งหมายเลขอาทิเช่นตัวอย่าง ข้างต้นเป็น port เบอร์ 6013 ซึ่งจะทำงานบนเครื่อง Server และรอว่ามี Client ต้องการร้องขอการเชื่อมต่อติดต่อเข้ามาหรือไม่ เราจะเห็นการเชื่อมต่อรูปแบบ Socket บ่อยครั้งในการเชื่อมต่อแบบ peer node ในระบบเครือข่ายของ Blockchain

1. **หากรัน DateClient ก่อนการรัน DateServer ได้ผลลัพธ์อย่างไร**

หากรัน Client ก่อน จะทำให้เกิดผลลัพธ์ดังภาพ



DateClient มีความพยายามที่จะเชื่อมต่อที่ port 6013 แต่ที่ port นั้นไม่ได้มีการถูกเปิดไว้ไม่มี process ที่ port 6013และหลังจากรัน DateServer ข้อความ ConectionException จะหายไปและถ้าหากทำการรันที่ DateClient อีกครั้งก็จะมีการ print ข้อความออกมา การทำงานสมบูรณ์เป็นปกติ DateServer จะต้อง listen connection ก่อนจะมีการ make connection จาก DateClient