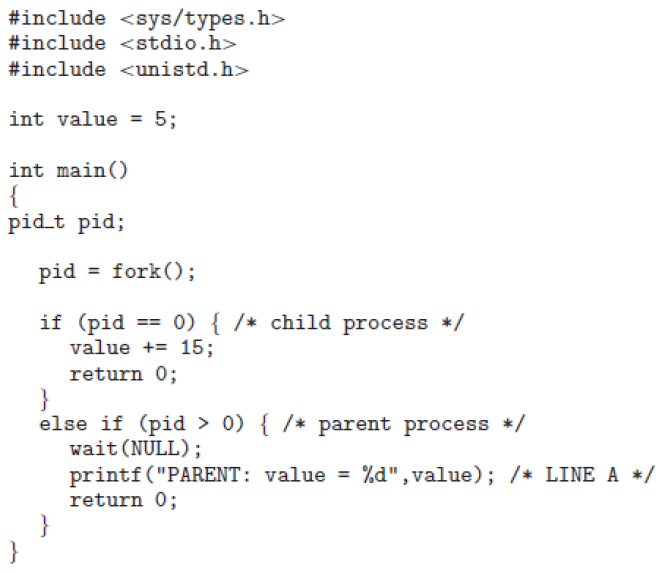
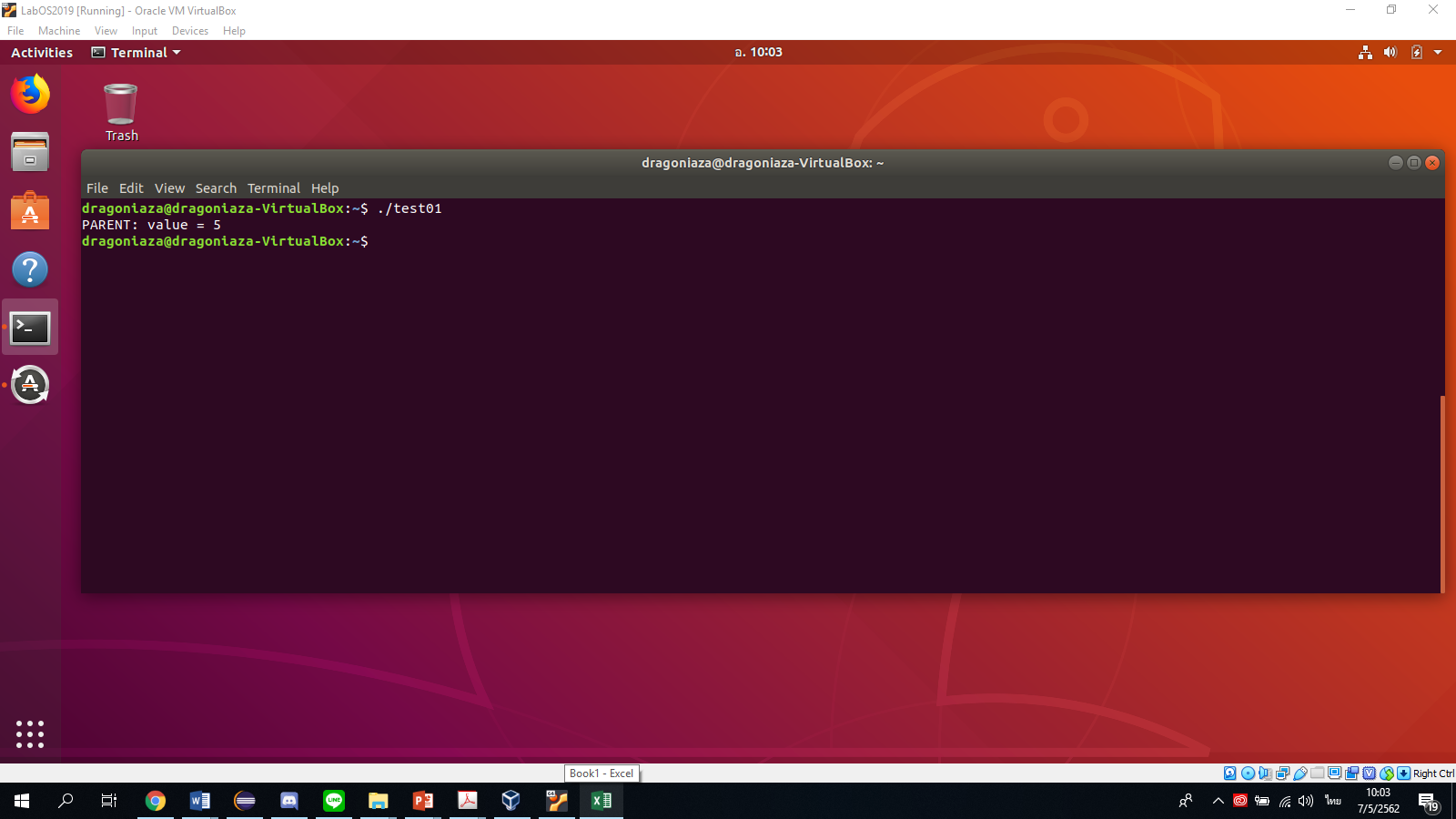
**แบบฝึกหัด บทที่ 3**

**1. จากโค้ดภาษาซีด้านล่างต่อไปนี้ โปรแกรมที่บรรทัด Line A ให้เอาท์พุต เป็นค่าใด จงทดลองและอธิบายผลการทดลอง**

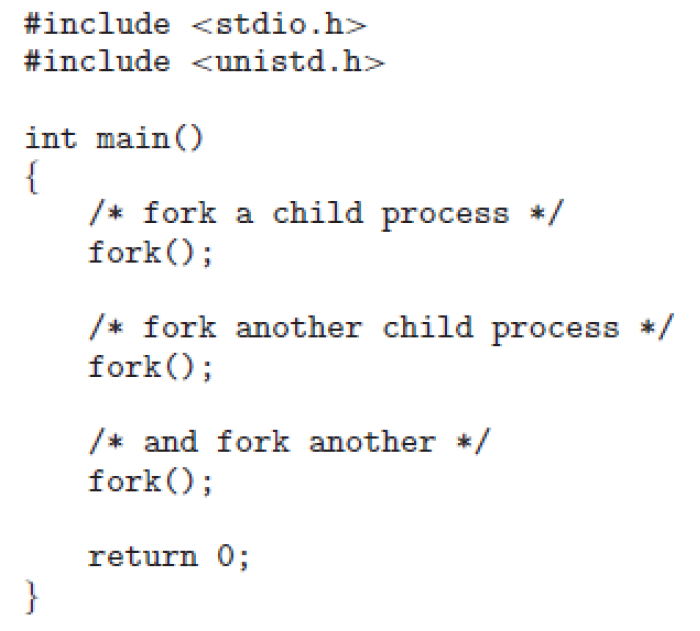




(ผลการรันโปรแกรม)

ในบรรทัดที่ A ให้ผลเป็นค่าของ parent process = 5 เพราะเนื่องด้วย โปรแกรมได้ทำการ fork parent process ขึ้นมาโดยที่จะตรวจสอบ id หากเป็น 0 ก็จะทำการ fork เพราะเป็น child process แต่หากไม่ใช่ก็จะรอและประกาศค่าของ parent process

**2. จากโค้ดภาษาซีต่อไปนี้ มีโปรเซสทั้งหมดกี่โปรเซส (นับรวมโปรเซส เริ่มต้นด้วย) จงอธิบาย**



8 Process โดยเริ่มต้น จะมี Parent process เพียง 1 เมื่อ fork ครั้งแรกจะเพิ่มจาก 1 เป็น 2 process จนได้ Child process เพิ่มเท่ากับ 2 และ Fork อีกครั้ง จะได้ Child process เพิ่มมาอีก 3 ตัวเป็น 4 และ fork ครั้งสุดท้ายจะได้ child เพิ่มมาอีก 4 จึงเท่ากับ 8

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

Shared Memory Segment เนื่องด้วย Heap นั้นเป็นการจองพื้นที่หน่วยความจำซึ่งเป็นแบบ Dynamic จริงอยู่ที่คุณสมบัติของโปรเซสแม่และลูกจะเหมือนกันทุกประการแต่ทั้งนี้ จะต้องใช้พื้นที่แตกต่างกัน จึงไม่สามารถแชร์สถานะนี้ได้ และ ส่วนของ Stack จะเก็บ local variable ซึ่งจะต้องถูกเก็บในทีต่างกัน

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

Short-term คือการที่หน่วยความจำได้คัดเลือกงานที่พร้อมจะ excecute และส่งต่อยังให้กับหน่วยประมวลผล

Meduim-term คือ ตัวจัดการตารางระยะกลาง คือ ย้าย Process ออกจากหน่วยความจำหลักเพื่อลดจำนวน Process ที่มีมากเกินไปในหน่วยความจำ เพื่อทำให้ CPU ทำงานได้ดีและเร็วขึ้น

Long-term คือ การที่คัดเลือกงานเพื่อนำเข้ามาสู่หน่วยความจำเพื่อเข้าสู่กระบวนการต่างๆ

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

โดยปกติแล้วนั้น จะบันทึก state สำหรับโปรเซสที่กำลังทำงานอยู่ ณ ขณะนั้น และจะทำการคืน state ให้กับโปรเซสตามที่ได้จัดไว้เพื่อให้ทำงานต่อไป การบันทึก state จะรวมถึงค่าของ register CPU ในการจัดสรรหน่วยความจำ

Context switch ยังคทำงานกับหล่ยๆสถาปัตยกรรมเฉพาะ รวมถึงการ flushing ข้อมูล และ instruction caches

**6. โปรเซส init ในระบบปฏิบัติการ Unix หรือ Linux คืออะไร มีหน้าที่อย่างไร จงอธิบายพอสังเขป**

เป็น process เริ่มต้นของ Linux จะทำงานเมื่อสั่งเปิดใช้ระบบปฏิบัติการโดยจะทำหน้าที่ Run level program และ Run shell เพื่อรอรับคั่งจากผู้ใช้

**7. Zombie Process และ Orphan Process คืออะไร เกิดขึ้นได้เมื่อใด จงอธิบาย**

Zombie process คือ โปรเซสที่ไม่ได้ทำงานแต่ยังกินทรัพยากรอยู่ เกิดจากการที่ Child ทำงานเสร็จก่อนและเข้าสู่สถานะ Terminate แต่ตัว Parent ยังไม่เสร็จสิ้นและได้เข้าสู่สถานะ waiting เพื่อรอการทำงานของ child โดยที่เราจะไม่สามารถลบหรือ เรียกให้ parent ทำงานได้อีก

Orphan Process คือ โปรเซส Child ยังทำงานไม่ได้ถูก execute แต่ตัว Parent ได้ terminate ตัวเองออกไปแล้ว ทำให้ Child เกิดการ Cascading คือ Child จะคงอยูในระบบแต่เรายังสามารถลบ Child ออกไปได้

**8. จากเรื่อง Interprocess Communication จงเขียนโปรแกรมเพื่อคำนวณ Collatz Conjecture จากอินพุต n ที่มีค่าบวกใดๆ โดย n ตัวใหม่คำนวณได้ดังต่อไปนี้ จนกว่า n จะมีค่าเป็น 1**

**n = 3\*n + 1 if n is odd**

**n = n/2 if n is even**

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

**ให้เขียนโปรแกรมโดยใช้ภาษาซี และใช้ POSIX Shared Memory ในโปรเซสหลักมีขั้นตอนดังนี้**

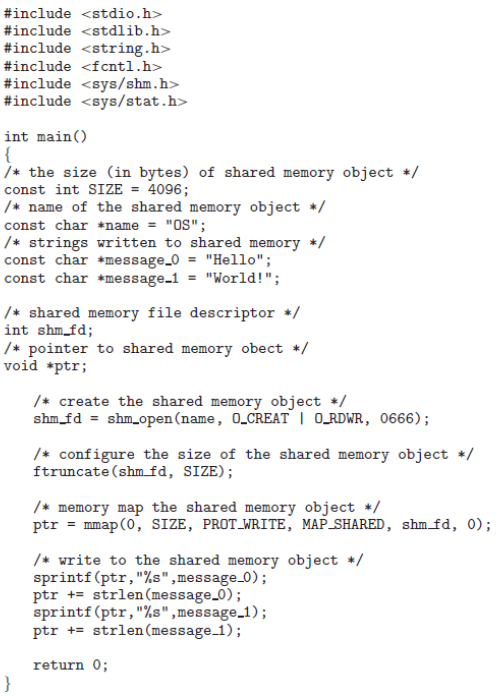
**1. สร้าง Shared-Memory Object (shm open(), ftruncate(), and mmap())**

**2. สร้างโปรเซสลูกเพื่อคำนวณ n โดยที่โปรเซสลูกเขียนค่า n ที่คำนวณได้ในแต่ละรอบ ลงใน Shared-Memory Object**

**3. รอจนโปรเซสลูกเสร็จ แล้วแสดงผลข้อมูลจาก Shared-Memory Object ออกทาง จอภาพ**

**4. ลบ Shared-Memory Object**

**ตัวอย่างของการใช้ Shared-Memory Object**



#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/mman.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/types.h>

#include "shared.h"

static int compute\_next\_term(int previous){

if(previous % 2 == 0){

return previous / 2;

}else{

return 3 \* previous + 1;

}

}

static void print\_sequence(int \*sequence, size\_t sequence\_length){

for(size\_t i = 0; i < sequence\_length; ++i){

printf("term: %d\n",sequence[i]);

}

}

int main(int arg\_count, const char \*\*args){

int term = atoi(args[1]);

int fd = shm\_open(SHM\_NAME, O\_CREAT | O\_RDWR, S\_IRWXU | S\_IRWXG | S\_IRWXO );

size\_t size = (MAX\_SEQUENCE\_LENGTH+1) \* sizeof(int);

ftruncate(fd, size);

int \*sequence = mmap(0, size, PROT\_WRITE, MAP\_SHARED, fd, 0);

sequence[0] = term;

size\_t sequence\_length = 1;

do{

term = compute\_next\_term(term);

sequence[sequence\_length] = term;

++sequence\_length;

}while(sequence\_length != MAX\_SEQUENCE\_LENGTH && term != 1);

sequence[sequence\_length] = STOP\_TOKEN;

print\_sequence(sequence, sequence\_length);

close(fd);

return 0;

* 1. }
  2. **(โปรแกรม Producer.c ในการทำ Collatz โดยที่จะเป็นตัว Writer ข้อมูลลงไปใน memory)**

#include <stdio.h>

#include <stdlib.h>

#include <sys/mman.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include "shared.h"

int main(int arg\_count, const char \*\*args){

int fd = shm\_open(SHM\_NAME, O\_RDONLY, S\_IRWXU | S\_IRWXG | S\_IRWXO);

size\_t size = (MAX\_SEQUENCE\_LENGTH + 1) \* sizeof(int);

int \*sequence = mmap(0, size, PROT\_READ, MAP\_SHARED, fd, 0);

while(\*sequence != STOP\_TOKEN){

printf("term: %d\n", \*sequence);

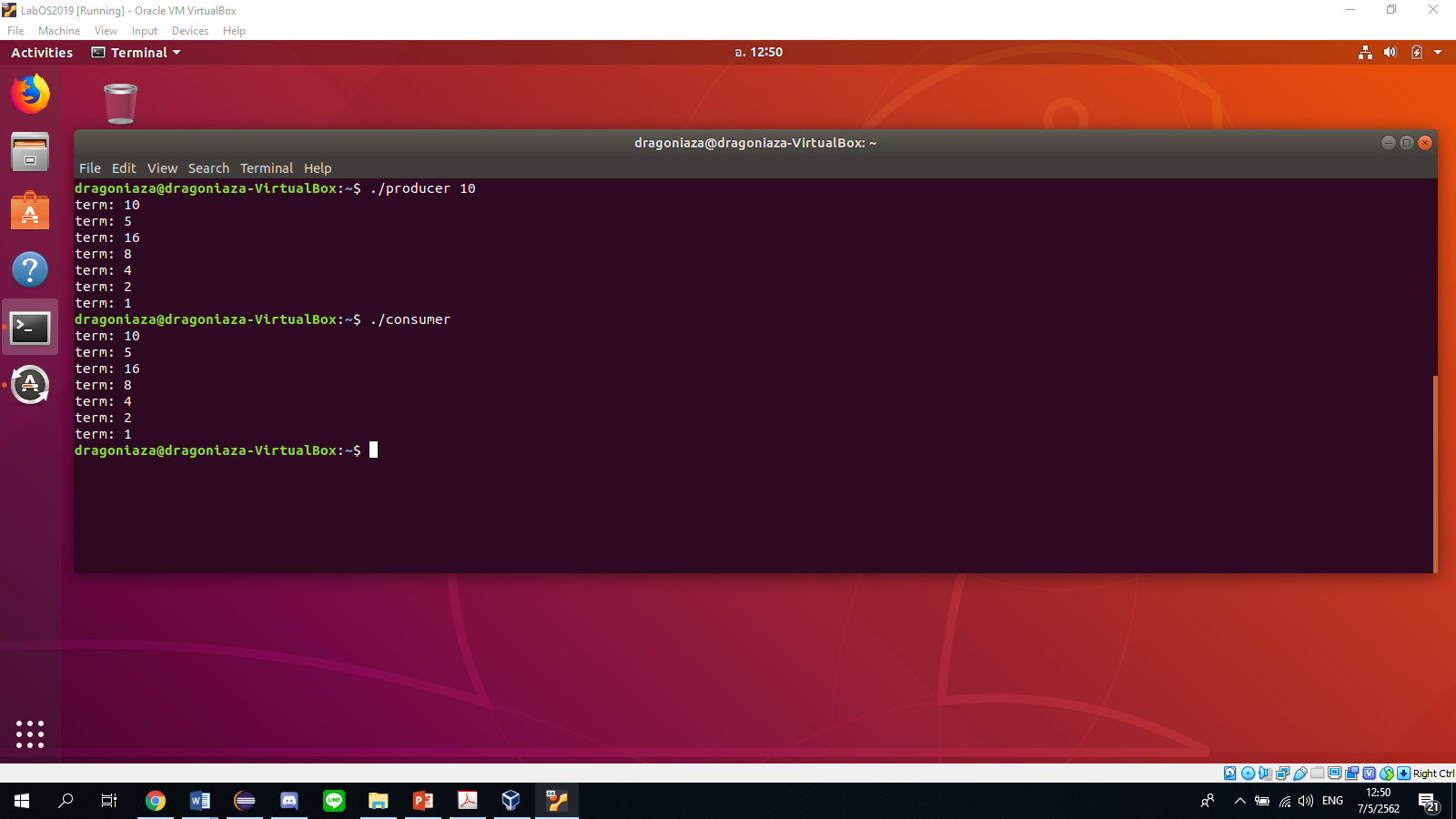
++sequence;

}

close(fd);

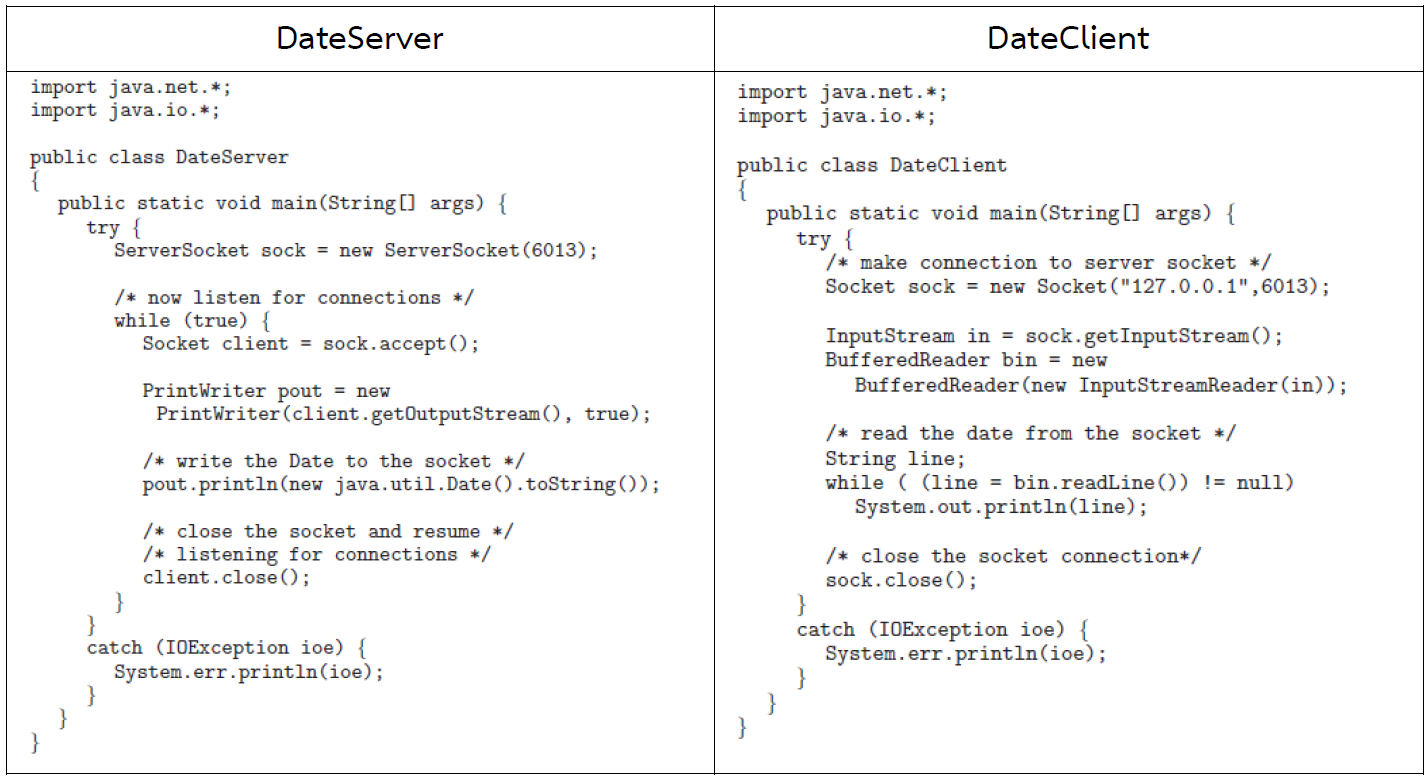
return 0;

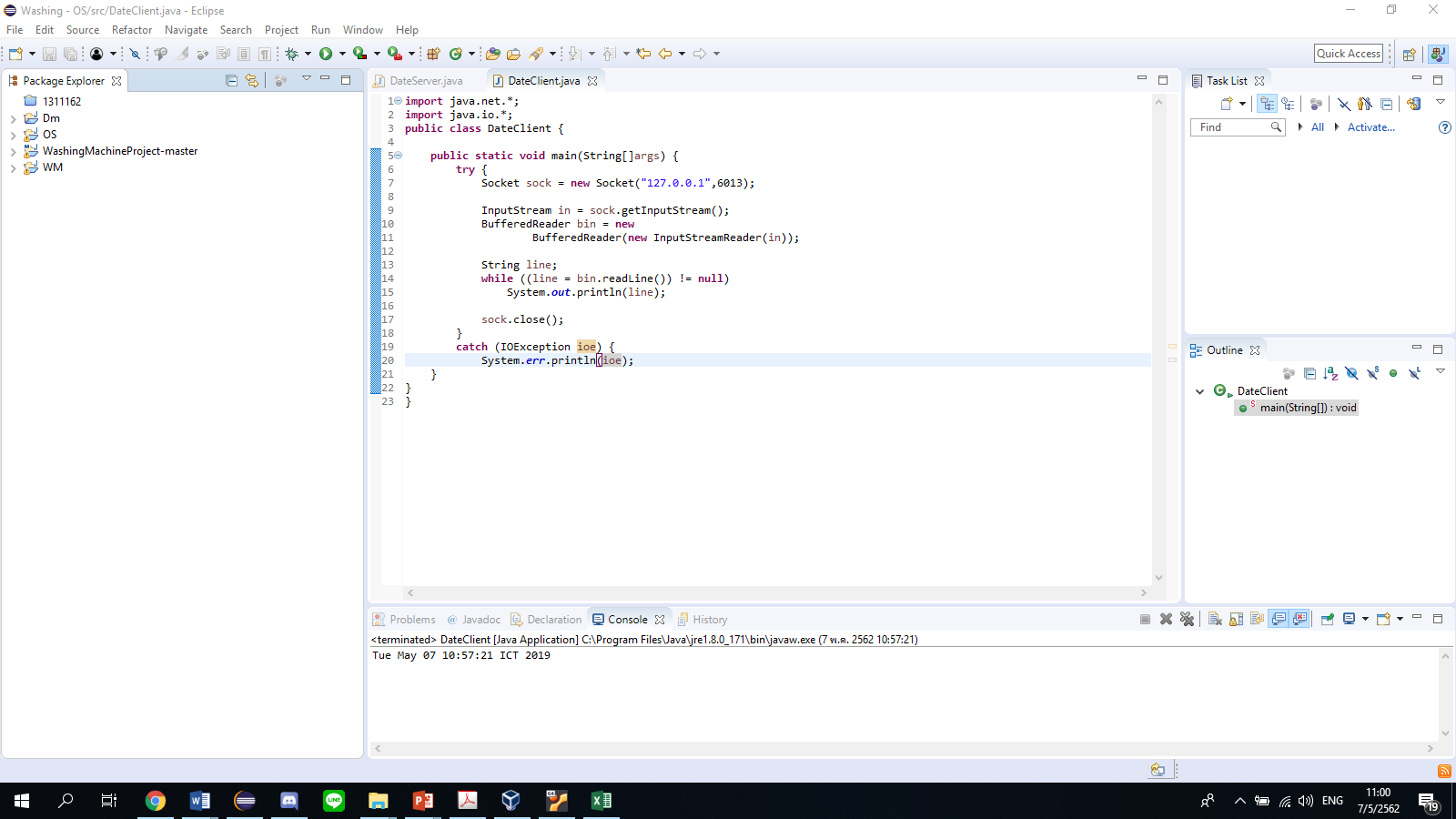
* 1. }
  2. **(โปรแกรม Consumer.c ที่เป็นตัว Read memory โดยดึงเอาข้อมูลจาก Producer มา)**



(ผลการรันโปรแกรม)

จะได้การรันของ **Collatz Conjecture** โดยที่เมื่อเสร็จสิ้นกระบวนการแล้วจะบันทึกข้อมูลลงใน Memory โดยที่จะเริ่มจากค่า พารามิเตอร์ที่เราใส่เข้าไป (อาทิ 10) แล้วจะผ่านการคิดตามเงื่อนใขต่างๆจนมีจำนวนเหลือแค่ 1 แล้วจากนั้นจะรัน consumer เพื่ออ่านค่าจากตัว memory ซึ่งจะต้องตรงกับคำตอบในตอนแรก

* 1. **9. จากเรื่อง Interprocess Communication จงทดลองโปรแกรมภาษาจาวา ต่อไปนี้ (หมายเหตุ ให้รัน DateServer ก่อน แล้วจึงค่อยรัน DateClient)**
  2. **a. โค้ดตัวอย่างข้างต้น มี Interprocess Communication ด้วยวิธีการใด**
  3. **b. การรันโปรแกรมได้ผลลัพธ์อะไร จงอธิบายพฤติกรรมของโปรแกรมอย่าง ละเอียด**
  4. **c. จงอธิบายความหมายของ Socket**
  5. **d. หากรัน DateClient ก่อนการรัน DateServer ได้ผลลัพธ์อย่างไร**
  6. 
  7. a.ตอบ ด้วยวิธีการแบบ Socket

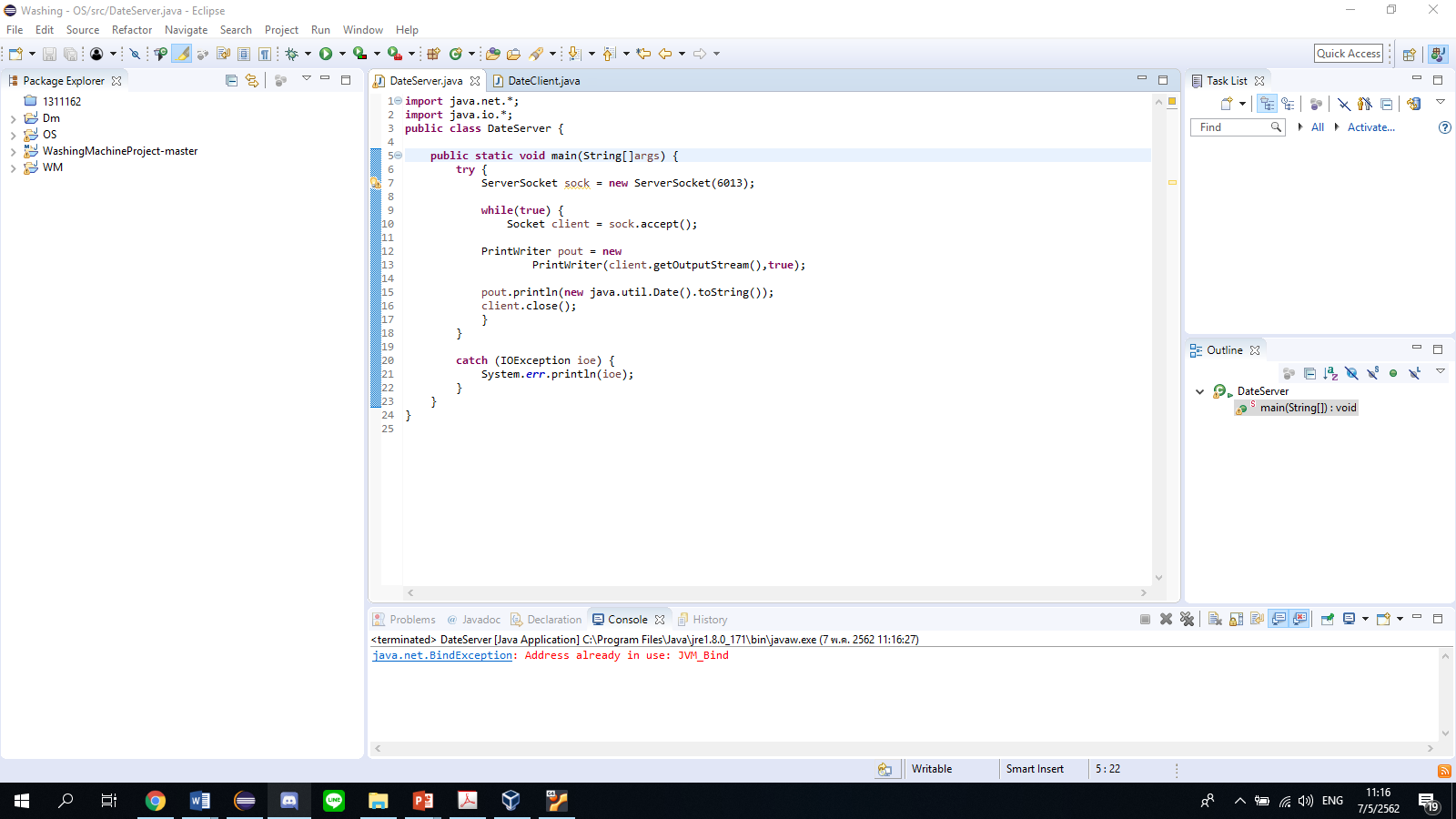
b. ผลลัพธ์คือการแสดงวันเวลาของตัว Client ที่ทำ

การเชื่อมต่อเข้ามายัง Port นั้น

b.(ต่อ) ในส่วนของ DateServer นั้นจะเป็นตัว Server ของการทดลองครั้งนี้ โดยได้เปิด port หมายเลข 6013 ไว้เพื่อรอการเชื่อมต่อและหากมีการเชื่อมต่อเข้ามาจะทำการแสดงผล วันที่และเวลา

ในส่วนของ DateClient จะทำการจำลองตัว Client ที่ได้เชื่อมต่อเข้าไปโดยได้เชื่อมต่อผ่าน loopback ที่ port6013 จากนั้นเมื่อเชื่อมต่อสำหรับก็จะรับค่า วันที่และเวลาจาก Socket ของ Server แล้วจึงจบการเชื่อมต่อ

c.เป็นการสื่อสาร ระหว่างอุปกรณ์ต่างชนิดกัน โดยจะเป็นตัวเชื่อมต่อระหว่าง IP และ Port โดยที่จะนับรวมแพคเกจของข้อความของ network service ที่แตกต่างกันจาก host ต่างๆ  
ซึ่งจะมีลักษณะการเชื่อมต่อโดยอาศัย Port เข้ามาเป็นตัวกำหนดว่าควรจะเชื่อมต่อผ่านช่องทางไหน

d.

(ผลการรันโปรแกรม)

เกิด error exception ขึ้นเนื่องด้วยตัว Client พยายามเชื่อมต่อกับ Server แต่ไม่สำเร็จเพราะยังไม่ได้เปิด port ดังกล่าวก็จะไม่สามารถ make connection ได้ส่งผลให้เกิดข้อความ exception ขึ้น