CPE 330 Operating Systems - Midterm Examination

2

ชื่อ _____ 27 ต.ค. 2553 _____

ส่วนที่ 1 เลือกข้อที่ถูกที่สุด (16 คะแนน ข้อละคะแนน) ในแต่ละคำถามจะมีคำตอบที่ถูกอยู่เพียงหนึ่งคำตอบ คุณสามารถเลือกทำสิ่งต่างๆ เหล่านี้ได้ในแต่ละข้อ

1. ปล่อยให้ว่าง – ไม่ได้คะแนน ไม่เสียคะแนน
2. เลือกหนึ่งคำตอบ – หากถูกจะได้ 1 คะแนน หากผิดจะถูกหัก 0.25 คะแนน
3. เลือกสองคำตอบ – หากถูกจะได้ 0.5 คะแนน หากผิดจะถูกหัก 0.5 คะแนน
4. เลือกมากกว่าสองคำตอบ – ไม่ได้คะแนน ไม่เสียคะแนน
5. คะแนนต่ำที่สุดในส่วนนี้คือ -1 คะแนนสูงที่สุดในส่วนนี้คือ 15

ทำเครื่องหมาย X ในช่องที่เหมาะสม																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a																
b																
c																
d																
e																
f																

1. มุมมองใดที่เกี่ยวข้องกับหน้าที่ของระบบปฏิบัติการผิดโดยสิ้นเชิง
   a. เป็นตัวจัดการทรัพยากรต่างๆ ของระบบคอมพิวเตอร์
   b. เป็นผู้จัดการสิทธิ์ในการใช้งานของผู้ใช้งานแต่ละคน
   c. เป็นตัวจัดการการทำงานของเธรดต่างๆ
   d. ตรวจสอบข้อผิดพลาดทางตรรกะของโปรแกรม (bug) และรายงานแก่ผู้ใช้ระบบ
   e. ทำตัวเป็นเสมือน virtual machine ให้โปรเซสใช้งาน
   f. ปิดบังความยุ่งยากซับซ้อนของฮาร์ดแวร์เพื่ออำนวยความสะดวกแก่ผู้พัฒนาโปรแกรม

2. การทำงานอะไรต่อไปนี้ไม่ควรเป็น system call ในการออกแบบระบบปฏิบัติการทั่วไป
   a. ขอหยุดการทำงานของโปรเซสชั่วคราว
   b. ขอเนื้อที่หน่วยความจำ (heap) เพิ่ม
   c. จัดการกับสัญญาณ interrupt ที่เข้ามา
   d. เปิดไฟล์เพื่ออ่านข้อมูลอย่างเดียว
   e. เขียนข้อมูลลงไฟล์
   f. สร้างหน่วยความจำร่วม (shared memory) เพื่อติดต่อกับโปรเซสอื่นๆ

3. ข้อความใดที่เกี่ยวกับ user/kernel mode ไม่ถูกต้อง
   a. ใน kernel mode จะประมวลผลได้เร็วกว่าใน user mode เพราะเป็น privilege mode
   b. ในการ context switch (ตั้งแต่ต้นจนจบ) จะต้องเกิดการเปลี่ยนโหมดระหว่าง user/kernel
   c. ใน user mode โปรเซสไม่สามารถอ่านค่าที่ถูกเก็บใน interrupt vector ได้
   d. คำสั่งที่อยู่ใน interrupt service routine จะต้องถูกประมวลผลใน kernel mode เสมอ
   e. System call จะถูกเรียกเมื่อโปรเซสทำงานอยู่ใน user mode เท่านั้น
   f. โปรเซสเซอร์อาจจะมีโหมดที่เกี่ยวข้องกับ privilege ในการประมวลคำสั่งได้มากกว่า 2 โหมดแต่ระบบปฏิบัติการสามารถที่จะเลือกใช้เพียงบางโหมดก็ได้

4. ข้อความใดที่เกี่ยวกับโครงสร้างของ kernel ผิด
   I. MS-DOS เป็น monolithic kernel
   II. Windows NT มีโครงสร้าง kernel แบบ hybrid kernel
   III. ถ้า kernel แบบ monolithic crash ระบบก็ crash ด้วย
   IV. ถ้า kernel แบบ microkernel crash ระบบก็ crash ด้วย
   a. I.
   b. II.
   c. III.
   d. IV.
   e. มีข้อที่ผิดมากกว่าหนึ่งข้อ
   f. ไม่มีข้อไหนผิดเลย

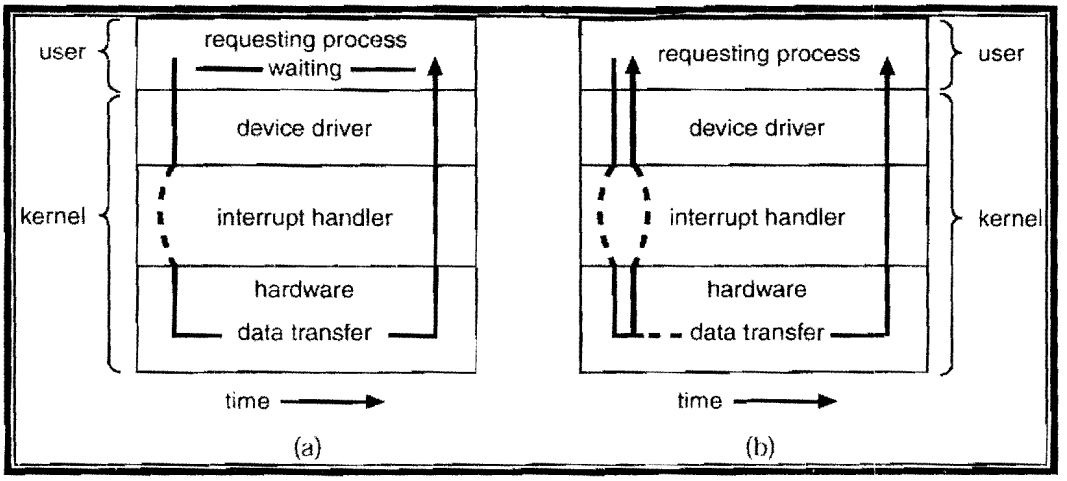
5. ขั้นตอนใดต่อไปนี้จะไม่มีทางเกิดขึ้นเมื่อเกิด hardware interrupt เข้ามาที่ CPU
   a. เปลี่ยนโหมดจาก user มา kernel โหมด
   b. ประมวลผล interrupt service routine ของทุกๆ device ที่ใช้ IRQ ที่เกิด interrupt ขึ้น
   c. kernel disable interrupt อื่นที่อาจจะเข้ามาในเวลาที่ยังอยู่ใน interrupt service routine
   d. เรียก scheduler และ dispatcher เพื่อเลือกโปรเซสที่ควรจะได้ใช้ CPU เพื่อทำงานต่อ
   e. PIC ส่งสัญญาณ acknowledge ให้กับ CPU เพื่อแสดงว่า interrupt ได้ถูกจัดการแล้ว
   f. ตรวจดูค่า address ของ interrupt service routine จาก interrupt vector

6. ข้อความใดที่เกี่ยวกับ virtual machine ไม่ถูกต้อง
   a. VMWare เป็นกลุ่มโปรแกรมที่เอาไว้จัดการ virtual machine
   b. ในการทำงานภายใต้สภาวะแวดล้อมแบบ virtual machine อาจจะมีเพียง 1 virtual machine ทำงานอยู่ก็ได้
   c. การทำ virtualization และการทำ emulation นั้นล้วนเป็นการจำลองฮาร์ดแวร์เหมือนกัน และมีประสิทธิภาพใกล้เคียงกัน
   d. การใช้ virtual machine เป็นการทำให้ทรัพยากรคอมพิวเตอร์ได้สามารถถูกใช้งานได้อย่างมีประสิทธิภาพมากขึ้น
   e. Guest OS ต่างๆ ในแต่ละ virtual machine อาจจะเป็นระบบเดียวกันหรือคนละตัวก็ได้ นอกจากนั้นยังอาจจะเป็น virtual machine เองก็ได้
   f. ปัจจุบันโปรเซสเซอร์บางกลุ่มเช่นพวก server ได้ถูกออกแบบมาให้รองรับการทำงานในสภาวะแวดล้อมแบบ virtual machine

7. ข้อความใดที่เกี่ยวกับโปรเซสกับเธรดไม่ถูกต้อง
   a. โปรเซสและเธรดต่างมี control block สำหรับเก็บข้อมูลของโปรเซสและเธรด
   b. หาก task มี 3 threads จะมี stack และ heap 3 ชุดสำหรับแต่ละ thread แยกจากกันชัดเจน
   c. หนึ่งโปรเซสอาจจะมีเพียงหนึ่งเธรดก็ได้
   d. โปรเซสใน Unix/Linux จะถูกสร้างขึ้นมาใหม่โดยใช้ system call fork()
   e. ในแต่ละ task จะมีการเก็บข้อมูลการใช้ทรัพยากรต่างๆ เช่น ไฟล์ ซึ่งอาจจะถูกใช้จากเธรดใดๆ ของ task นั้นก็ได้
   f. เธรด T_1 ของโปรเซส P_1 และเธรด T_2 ของโปรเซส P_2 ไม่สามารถอ่านเขียนข้อมูลร่วมกันได้ ถ้าไม่มีการทำ shared memory ขึ้นมาต่างหาก

รหัสนักศึกษา _____

8. จากรูปด้านล่าง ข้อใดถูก



I. รูป a เป็นการทำงานแบบ synchronous รูป b เป็นการทำงานแบบ asynchronous

II. รูป a เป็นการทำ interrupt รูป b เป็นการทำ polling

III. หากระบบมีเพียงโปรเซสเดียวที่ทำงาน รูป b สามารถเพิ่มประสิทธิภาพการทำงานได้ เพราะทำงาน CPU และ I/O ไปพร้อมๆ กัน

IV. ระหว่างเกิด hardware data transfer โปรเซสในรูป a จะอยู่ใน waiting state แต่โปรเซสในรูป b จะอยู่ใน ready state

a. I, II.

b. I, III.

c. II, III.

d. I, II, III, IV.

e. II, IV.

f. III, IV.

9. ข้อความใดที่เกี่ยวกับ polling และ interrupt ผิด
   I. ถ้าอุปกรณ์ไอโอใช้เวลาสั้นมากในการทำงานที่ได้รับมอบหมาย การทำ polling จะดีกว่า interrupt เพราะไม่ต้องมาเสียเวลากับ interrupt service routine
   II. ในรูปแบบ interrupt เมื่ออุปกรณ์ไอโอทำงานเสร็จจะแจ้งให้ CPU ทราบผ่าน device controller โดยมีการ set ค่า IRQ
   III. หากการทำงานอยู่ใน kernel mode จะไม่มีการตรวจสอบว่ามี interrupt เข้ามาที่โปรเซสเซอร์หรือไม่ ก่อนที่จะ execute ทุกๆ คำสั่ง(ภาษาเครื่อง) เนื่องจาก interrupt จะถูก disable โดยอัตโนมัติ

   a. I.
   b. II.
   c. III.
   d. I, II.
   e. II, III
   f. I, III.

10. ข้อความใดที่เกี่ยวกับการ reentrant kernel ระบบผิด
   a. Reentrant kernel คือการที่ kernel สามารถให้มีโปรเซสมากกว่าหนึ่งโปรเซสทำงานอยู่ใน kernel mode ได้ในเวลาเดียวกัน
   b. การใช้ global variable ในส่วนของ user mode ไม่มีผลต่อการออกแบบและการทำงานของ reentrant kernel
   c. หากจำเป็นต้องใช้ global kernel variable ระบบที่เป็น reentrant kernel จะต้องทำการ lock การใช้ตัวแปรเหล่านั้นก่อน เพราะไม่ให้โปรเซสอื่นทำการแก้ไขระหว่างการใช้ตัวแปรดังกล่าว
   d. Reentrant kernel จะมีในระบบปฏิบัติการที่เป็น preemptive scheduling เท่านั้น
   e. การใช้ local kernel variable นั้นจะสอดคล้องกับหลักการของ reentrant kernel เนื่องจาก แต่ละ kernel execution path จะมี kernel stack เป็นของตนเองไม่ปนกับตัวอื่นๆ
   f. ไม่มีข้อความใดผิดเลย หรือ มีข้อความที่ผิดมากกว่าหนึ่งข้อ

[redacted]

11. ข้อใดต่อไปนี้ไม่ใช่สาเหตุของการเกิด context switch
    a. การที่มีโปรเซสที่มี priority ต่ำกว่าเข้ามาที่ ready queue
    b. การเรียกใช้ system call ที่ทำให้เกิดการรอเป็นระยะเวลานาน
    c. การที่โปรเซสทำงานจนหมด time slice ที่ได้รับ
    d. การเกิด exception ระหว่างการทำงาน
    e. การจบการทำงานของโปรเซส
    f. การเรียก wait() ของ semophore

12. ข้อความใดที่เกี่ยวกับการสร้าง การติดต่อและการจบ process ในระบบ UNIX ผิด
    a. ในการสร้างโปรเซส ถ้ามีการเรียก fork() ก่อน pipe() โปรเซสลูกจะไม่สามารถติดต่อกับโปรเซสแม่ผ่าน pipe ได้
    b. โปรเซส P1 สร้างลูกสองตัวคือ P2, P3. ตามหลักการแล้ว P2 และ P3 สามารถติดต่อกันผ่าน pipe ได้ถ้ามีการเรียก system call อย่างถูกต้อง
    c. ทันทีหลังจาก fork() โปรเซสลูกที่ถูกสร้างมาจะมีส่วน code, data, stack เหมือนโปรเซสแม่ทุกประการ เพียงแต่เป็นคนละโปรเซส โดยโปรเซสลูกจะเริ่มทำงานจากในคำสั่งที่ต่อจาก fork()
    d. หากโปรเซสลูกจบการทำงานก่อนโปรเซสแม่ โปรเซสลูกจะอยู่ในสถานะ zombie จนกว่าโปรเซสแม่จะจบการทำงาน
    e. หากโปรเซสแม่จบการทำงานก่อนโปรเซสลูก ระบบอาจจะจบการทำงานของโปรเซสลูกทุกๆ ตัว หรือโอนโปรเซสลูกทั้งหลายให้เป็นลูกบุญธรรมของโปรเซส 1
    f. ไม่มีข้อความใดผิดเลย หรือ มีข้อความที่ผิดมากกว่าหนึ่งข้อ

รหัสนักศึกษา _____

13. ข้อความใดที่เกี่ยวกับ thread ถูก

I. การสร้าง thread ทำได้เร็วกว่าการสร้างโปรเซสเพราะไม่ต้อง allocate เนื้อที่ในหน่วยความจำเลย

II. thread ที่อยู่ใน process เดียวกันใช้ code และ statically allocated data ร่วมกัน

III. แต่ละ thread จะมีค่าของ program counter ของมันเอง

IV. ใน M:1 thread model หากเธรดของโปรเซส P1 ถูก block เธรดอื่นๆ ของโปรเซส P1 ก็จะถูก block ด้วย

V. User-level thread library จะใช้ thread model แบบ M:N

a. I, III, V.

b. II, IV, V.

c. II, III, V.

d. I, III, IV, V.

e. II, III, IV.

f. I, II, IV.

14. จาก Bakery algorithm ด้านล่าง ข้อความใดไม่ถูกต้อง

```
1:  while(1) {
2:       choosing[i] = true;
3:       number[i] = max(number[0], number[1], …, number[n - 1])+1;
4:       choosing[i] = false;
5:       for (j = 0; j < n; j++) {
6:         while (choosing[j]);
7:         while ((number[j] != 0) && (number[j],j) < (number[i],i));
8:       }
9:         /* critical section */
10:      number[i] = 0;
11:        /* remainder section */
12: }
```

a. เราไม่สามารถตัดบรรทัดใดบรรทัดหนึ่งในส่วน entry/ exit code แล้วทำให้ algorithm ยังคงความถูกต้องได้

b. ทั้ง choosing[] และ number[] ต้องเป็นตัวแปร shared memory ทั้งคู่

c. เปลี่ยนบรรทัด 3 เป็น number[i] = 1; algorithm จะผิดพลาดเพราะสามารถทำให้เกิด unbounded waiting ได้

d. หากตัดบรรทัดที่ 6 จะสามารถทำให้เงื่อนไข progress ผิด

e. หากเงื่อนไข (number[j] != 0) ออกจากบรรทัดที่ 7 จะทำให้เงื่อนไข mutual exclusion ผิด

f. ไม่มีข้อความใดผิดเลย หรือ มีข้อความที่ผิดมากกว่าหนึ่งข้อ

15. จาก Synchronization algorithm โดยใช้ semaphore ข้างล่างนี้ ข้อความใดถูก

```
1:while(1) {
2:  waiting[i] = true;  key = true;
3:  while (waiting[i] && key)
4:        key = test_and_set(lock);
5:  waiting[i] = false;
6:        CS
7:  j = (i+1) % n;
8:  while (( j != i) && !waiting[j])
9:        j = (j + 1) % n;
10: if (j == i)
11:       lock = false;            // no other proc waits, release lock
12: else
13:       waiting[j] = false;      // set turn to proc j
14: remainder section
15:}
```

a. ทั้ง waiting[], key, lock เป็นตัวแปร shared memory

b. Algorithmนี้เป็นไปตามเงื่อนไข progress เพราะว่าถ้าไม่มีใครเข้า critical section ตัวแปร lock จะเป็น false ทำให้โปรเซสที่ต้องการเข้า critical section สามารถเข้าได้

c. Algorithm นี้ไม่สามารถใช้ร่วมกับ blocking semaphore ได้

d. หากไม่มีการตรวจสอบโปรเซสที่คอยในบรรทัดที่ 7-9 และทำเพียงบรรทัดที่ 11 จะทำให้ algorithm ผิดเงื่อนไข mutual exclusion และ bounding waiting

e. ในบรรทัดที่ 7-9 เราสามารถตรวจสอบการคอยของโปรเซสอื่นๆ ในลำดับอื่นๆ ได้ เช่นไล่จากโปรเซส (i-1)%n ลงไปจนวนกลับมาที่โปรเซส i แต่จะทำให้เกิดการผิดเงื่อนไข progress ได้

f. มีข้อความที่ถูกมากกว่าหนึ่งข้อ หรือไม่มีข้อความที่ถูกเลย

16. ข้อความใดต่อไปนี้ที่เกี่ยวกับ semaphore ผิด

a. Binary semaphore คือ semaphore ที่มีค่า 0 และ 1 เท่านั้น ไม่สามารถติดลบหรือมีค่ามากกว่านั้นได้

b. การใช้ semaphore สามารถทำให้เกิด deadlock ในระบบได้

c. ทั้ง Windows และ Linux มี semaphore ให้สำหรับผู้ที่ต้องการทำการ synchronization

d. Semaphore สามารถป้องกัน race condition ได้หากใช้อย่างถูกต้อง

e. โปรเซสสามารถใช้ semaphore หลายๆ ตัวพร้อมๆ กันได้

f. ในกรณีที่การเรียก wait() ทำให้โปรเซสเปลี่ยนไปคอยใน waiting state โปรเซสจะเข้าไปอยู่ใน waiting queue ของ semaphore ตัวนั้น

[redacted]

รหัสนักศึกษา _____

ส่วนที่ 2 ตอบคำถาม

1. จาก find out more ใน lecture และจาก OSP2 project
   a. Windows มีตัวจัดการ multi-boot เพื่อเลือก Windows image ของ version ที่ต้องการได้ (เช่นลงทั้ง XP, Vista จะสามารถเลือก boot ได้) ไฟล์อะไรที่เกี่ยวข้องกับ multi-boot นี้ และมีขั้นตอนคร่าวๆ ในการกำหนด multi-boot อย่างไร (1 คะแนน)

   b. อะไรคือ deadlock avoidance และต่างจาก deadlock prevention อย่างไร (1 คะแนน)

   c. ใน OSP2 ThreadCB เป็น class ที่ inherit มาจาก class ชื่ออะไร และ ThreadCB ควรจะเก็บข้อมูลอะไรบ้าง (1 คะแนน)

[redacted]

รหัสนักศึกษา _____

2. จงขีดเส้นของข้อความที่มีความหมายผิด (รวมถึงที่ผิดทั้งหมด และ ผิดบางส่วน) โดยข้อความแต่ละข้อความจะถูกแยกโดยเครื่องหมาย / (ข้อละ 2 คะแนน)

   I. ปัญหา dinning philosopher เป็นปัญหาทางด้าน synchronization ปัญหาหนึ่งซึ่งในกรณีที่ได้พูดใน class จะมีโปรเซสอยู่ 5 ตัวเรียงกันเป็นวงกลม และมี resource อยู่ 5 หน่วยกระจายระหว่างโปรเซสที่ติดกัน / แต่ละโปรเซสจะมี critical section และ remainder section โดย critical section จะทำการใช้ resource 2 ตัวที่อยู่ติดกับตนเอง และ remainder section จะใช้ resource เพียง 1 ตัวเท่านั้น / หากมีโปรเซสมากกว่า 2 โปรเซสอยู่ใน cr tical section ในเวลาเดียวกัน จะทำให้เกิด deadlock อย่างแน่นอนไม่ว่าลำดับการทำงานจะเป็นอย่างไร (สมมุติว่าโปรเซสอื่นๆ ไม่เข้า critical section ในเวลาช่วงนั้น) / หากมีโปรเซส 2 โปรเซสอยู่ใน critical section อาจจะเกิด deadlock ได้ในกรณีที่เป็นโปรเซสติดกัน / แต่ถ้ามีเพียงโปรเซสเดียวที่อยู่ใน critical section จะไม่มีทางเกิด deadlock ได้

   II. test_and_set() เป็น function พิเศษของระบบปฏิบัติการที่ทำงานใน kernel mode เท่านั้น / โดยมีหลักการว่าการทำงานทั้งหมดจะเป็น atomic / ซึ่งการทำงานแบบ atomic หมายถึงการที่ไม่สามารถมีการแทรกการทำงานโดยโปรเซสอื่นระหว่างกึ่งกลางของการทำงาน test_and_set() ได้ / ทันทีหลังจากการเรียก test_and_set(v) ตัวแปร v จะถูก set ค่าเป็น true (1) ส่วนค่าที่ return จาก test_and_set() จะเป็นค่าเก่าของตัวแปร v / ซึ่ง test_and_set() นี้สามารถใช้แทน semaphore ได้ในทุกกรณี เพราะป้องกัน critical section เหมือนกัน

3. กำหนดให้ workload ในการทำงานเหมือนกัน จงเรียงลำดับของ thread model (x:y) ตามเงื่อนไขต่อไปนี้ (1 คะแนน)

   I. การถูก block ของ thread จากน้อยไปมาก

   II. จำนวน kernel level thread จากน้อยไปมาก

รหัสนักศึกษา _____

4. กำหนดให้ time slice = 10 units, context switch เสียเวลาครั้งละ 1 unit (แต่ถ้าเรียก scheduler เฉยๆ ไม่เสียเวลา) และมี process burst ดังนี้

Process	Arrival time	Burst
P0	0	42
P1	5	8
P2	23	30
P3	49	16

a. วาด Gantt chart ของการ schedule โปรเซสเมื่อใช้ round-robin algorithm พร้อมหาค่า average waiting time (1 คะแนน)

b. กำหนดให้ time slice ของ 1^{st}-level queue = 10 หน่วย และ time slice ของ 2^{nd}-level queue = 20 หน่วย วาด Gantt chart ของการ schedule โปรเซสเมื่อใช้ multi-level feedback queue algorithm (แต่ละ level เป็น round robin) พร้อมหาค่า average waiting time(1 คะแนน)