ที่นั่งสอบ _____

King Mongkut's University of Technology Thonburi

Midterm Exam of Second Semester, Academic Year 2007

COURSE CPE 330 Operating Systems **Computer Engineering Department, 3rd Yr.**

Wednesday 19 December 2007 **9.00-12.00**

Instructions

1. This examination contains 3 parts, 19 problems, 12 pages (including this cover page).
2. The answers must be written in this exam paper. For multiple choice problems, mark the appropriate choice(s) in the table provided. For other problems, write the answer in the space provided.
3. No books, notes, or any other documents, calculator can be taken into the examination room.

Students will be punished if they violate any examination rules. The highest punishment is dismissal.

This examination is designed by
Dr. Songrit Maneewongvatana
Tel. 0-2470-9083

รหัสนักศึกษา _____

CPE 330 Operating Systems - Midterm Examination

ชื่อ _____

ส่วนที่ 1 เลือกข้อที่ถูกที่สุด (12 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 12

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

1. ข้อใดไม่ใช้หน้าที่หรือความรับผิดชอบของระบบปฏิบัติการ
   a. จัดการทรัพยากรของระบบ
   b. ป้องกันไม่ให้โปรเซสหนึ่งไปเข้าถึงหน่วยความที่ถูกจัดสรรให้กับโปรเซสอื่นๆ
   c. สร้างโปรเซสใหม่
   d. คอยดักจับ virus เพื่อรักษาความปลอดภัยของระบบ
   e. จัดการสิทธิ์ในการใช้งานของผู้ใช้
   f. คอยตรวจสอบว่ามีอุปกรณ์เสริม (เช่น flash drive) มาเชื่อมต่อกับระบบหรือไม่

2. ข้อความใดที่เกี่ยวกับ user/kernel mode ไม่ถูกต้อง
   a. Supervisor ของระบบ จะสามารถเรียกโปรเซสให้ทำงานในโหมด kernel ได้
   b. ในการทำงานใน user mode โปรเซสไม่สามารถประมวลผลคำสั่งไอโอได้
   c. เมื่อเกิด interrupt ขึ้น ไม่จำเป็นว่าจะต้องมีการเปลี่ยนโหมดเสมอไป
   d. System call จะถูกเรียกเมื่อโปรเซสอยู่ใน user mode เท่านั้น
   e. Kernel mode, supervisor mode, admin mode คือโหมดเดียวกัน
   f. คุณสมบัติ user/kernel mode นี้เป็นคุณสมบัติของโปรเซสเซอร์ ระบบปฏิบัติการเป็นแค่ตัวใช้คุณสมบัตินี้ในการจัดการระบบเท่านั้น

3. ข้อความใดที่เกี่ยวกับ virtual machine ผิด
   a. VM Ware เป็นหนึ่งในโปรแกรมที่เอาไว้ใช้ทำ virtual machine
   b. เป็นไปได้ที่ host OS และ guest OS จะเป็นระบบปฏิบัติการเดียวกัน
   c. ในระบบหนึ่งอาจจะมี host OS และ guest OS หลายๆ ตัว ทำงานพร้อมกันได้
   d. Guest OS จะมองไม่เห็นทรัพยากรต่างๆ (เช่น ไฟล์) ที่ guest OS ตัวอื่นเป็นเจ้าของอยู่
   e. จุดประสงค์หลักของ JVM หรือ Java Virtual Machine จะแตกต่างจาก virtual machine ปกติ
   f. หาก host OS เกิดมีปัญหาแฮงค์ขึ้นมา guest OS ก็จะไม่สามารถทำงานต่อได้
   g. ข้อดีประการหนึ่งของการทำ virtual machine คือเป็นการแบ่งกันใช้ทรัพยากรฮาร์ดแวร์ที่มีอยู่อย่างจำกัดโดยระบบปฏิบัติการหลายๆ ตัว

4. ขั้นตอนใดต่อไปนี้ไม่เป็นขั้นตอนที่ระบบปฏิบัติการอาจจะทำเมื่อมี hardware interrupt เกิดขึ้น?
   I. บันทึก context ของโปรเซสที่กำลังทำงานอยู่
   II. ติดต่อ device driver ให้ส่ง IRQ ไปยังอุปกรณ์ไอโอเพื่อแจ้งให้รับทราบ
   III. Disable interrupt อื่นที่อาจจะเข้ามา
   IV. เรียก system call เพื่อทำจัดการกับ interrupt ที่เกิดขึ้น
   V. ดูค่าใน interrupt vector เพื่อหาตำแหน่งของ service routine ที่ถูกต้อง

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

5. ข้อความใดที่เกี่ยวกับ system call และพารามิเตอร์ของ system call ถูก
   a. ระบบปฏิบัติการสามารถส่งค่าพารามิเตอร์ผ่านรีจิสเตอร์ได้เสมอเนื่องจากค่าในรีจิสเตอร์จะถูกรักษาไว้เมื่อมีการเรียก system call
   b. การเรียก system จะทำให้โปรเซสเปลี่ยนสถานะของการทำงาน (running, ready, waiting) เสมอ
   c. การส่งค่าพารามิเตอร์ผ่าน stack เป็นวิธีที่เหมาะสมกับพารามิเตอร์ที่มีขนาดสั้นๆเท่านั้น
   d. หากส่งค่าพารามิเตอร์ผ่านตำแหน่งในหน่วยความจำ พารามิเตอร์สามารถมีขนาดใหญ่ได้
   e. ในการส่งค่าพารามิเตอร์ผ่าน stack โปรเซสจะส่งค่าพารามิเตอร์ไปที่ kernel-mode stack เพื่อให้ system call routine รับค่าได้โดยตรง
   f. ระบบปฏิบัติการสามารถใช้รีจิสเตอร์ตัวไหนเก็บค่าพารามิเตอร์ของ system call ก็ได้

6. เมื่อสถานะของโปรเซส P1 เปลี่ยนจาก waiting มาเป็น ready กรณีใดบ้างที่โปรเซส P1 จะไม่สามารถเปลี่ยนสถานะเป็น running ได้ทันที
   I. ready queue ว่าง
   II. P1 มี priority ต่ำกว่าโปรเซสที่ทำงานอยู่ และ scheduling algorithm เป็น priority scheduling ชนิดpreemptive
   III. P1 มี priority สูงกว่าโปรเซสที่ทำงานอยู่ และ scheduling algorithm เป็น priority scheduling ชนิด non-preemptive
   IV. P1 มี priority สูงกว่าโปรเซสอื่นๆ ที่อยู่ใน ready queue และ scheduling algorithm เป็น round robin

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

7. ข้อความใดที่เกี่ยวกับ polling และ interrupt ผิด
   a. โปรเซสไม่สามารถจะ poll อุปกรณ์ไอโอได้โดยตรง
   b. ถ้าอุปกรณ์ไอโอใช้เวลาสั้นในการทำงานที่ได้รับมอบหมาย การทำ interrupt จะดีกว่า polling เพราะจะรบกวน CPU น้อยกว่า
   c. ในรูปแบบ interrupt เมื่ออุปกรณ์ไอโอทำงานเสร็จจะแจ้งให้ CPU ทราบผ่าน device controller โดยมีการ set ค่า IRQ
   d. ในรูปแบบ interrupt จะสามารถทำให้เกิด context switch ได้ เมื่อมี IRQ
   e. การยอมให้มี nested interrupt ทำให้ระบบมีการตอบสนองต่ออุปกรณ์ไอโอที่ดีกว่าแต่จะมีความยุ่งยากเพิ่มขึ้นเนื่องจากสามารถมี interrupt service routine มากกว่าหนึ่ง routine ทำงานอยู่ใน kernel mode
   f. ในระบบปฏิบัติการตระกูล Windows NT มีการจัดการอุปกรณ์ไอโอโดยใช้ interrupt

8. ข้อความใดที่เกี่ยวกับการ boot ระบบ (เช่นเครื่อง PC) ผิด
   a. จะต้องมีการตรวจสอบอุปกรณ์สำคัญของระบบ ที่เรียกว่าPOST เสมอ
   b. ส่วนหนึ่งของ BIOS จะเก็บค่าพารามิเตอร์ของ hard disk เพื่อช่วยในการหาตำแหน่งของ boot sector
   c. Code ใน boot sector นั้นไม่สามารถจะโหลดตัว image ของ OS ทั่วไปได้ เนื่องจาก boot sector มีขนาดเล็กทำให้ไม่สามารถบรรจุรหัสได้เพียงพอ
   d. ในการทำ dual boot จะต้องกำหนดค่าการ boot ใน BIOS ว่าจะให้ boot จากระบบปฏิบัติการใดบ้าง
   e. ข้อมูลใน BIOS จะถูกเก็บในหน่วยความจำประเภทไม่สูญหายเมื่อดับไฟ เช่น flash memory
   f. ไม่มีข้อความใดผิดเลย หรือ มีข้อความที่ผิดมากกว่าหนึ่งข้อ

9. ข้อความที่เกี่ยวกับสถานะของโปรเซสต่อไปนี้ข้อความใดบ้างที่ถูก (กำหนดให้เป็น single core)

   I. จะมีเพียงโปรเซสเดียวเท่านั้นที่จะอยู่ในสถานะ running

   II. จะต้องมีโปรเซสอยู่ใน ready queue เสมอ

   III. หากโปรเซสเปลี่ยนจากสถานะ waiting ไปเป็น ready ตัว scheduler จะต้องถูกเรียกเสมอ

   IV. ใน non-preemptive scheduling จะไม่มีการเปลี่ยนสถานะจาก running ไปเป็น ready

   V. ในสถานะ terminated โปรเซสจะต้องเป็น zombie เสมอ

   a. I, II, V.

   b. II, III, IV.

   c. I.

   d. I, III, V.

   e. I, III.

   f. I, III, IV,

10. ข้อความใดที่เกี่ยวกับ process communication ผิด

   a. ถ้าเป็น blocking send โปรเซสที่ส่งข้อความจะต้องรอให้ผู้รับมารับเสมอ ไม่สามารถทำอย่างอื่นต่อเลยได้ถ้าผู้รับยังไม่รับข้อความ

   b. ถ้าการส่งเป็นแบบ non-blocking send / blocking receive ระบบจะต้องมี buffer มิฉะนั้นแล้วผู้รับจะไม่ได้รับข้อความ

   c. ถ้าต้องการรับส่งข้อความตามลำดับ FIFO ให้กับโปรเซส parent-child เราสามารถใช้ pipe หรือ named pipe ก็ได้

   d. ตัว IPC message จะยังไม่ถูกลบแม้ว่าโปรเซสที่ส่ง IPC message จบการทำงานและออกจากระบบแล้ว

   e. โปรเซสหลายๆโปรเซสในเครื่องเดียวกันสามารถเข้าถึงส่วน shared memory ที่สร้างจาก IPC shared memory ได้ โดยที่ไม่จำเป็นจะต้องเป็น parent-child หรือ ancestor-descendent

   f. ไม่มีข้อความใดผิดเลย หรือ มีข้อความที่ผิดมากกว่าหนึ่งข้อ

11. ข้อความใดที่เกี่ยวกับ thread ถูก
    a. การสร้างเธรดทำได้เร็วกว่า การสร้างโปรเซสใหม่ เนื่องจากเธรดมีขนาดเล็กกว่าโปรเซส
    b. ใน M:1 thread model เธรดของโปรเซส P1 สามารถ block การทำงานของ เธรดของโปรเซส P2 ได้ เนื่องจากทั้งสองเธรดจะ share kernel-level เธรดเดียวกัน
    c. เธรดในโปรเซสเดียวกันจะใช้ code, ข้อมูล และ stack ร่วมกัน
    d. หากระบบปฏิบัติการมี 1:1 thread model เราสามารถเปลี่ยนให้เป็นแบบ M:N ได้โดยการเพิ่ม user level thread library เข้าไปบนแต่ละ kernel-level เธรด
    e. การจัดการ kernel-level เธรดจะมีประสิทธิภาพน้อยกว่า user-level เธรด เนื่องจากจะต้องมีการทำ mode switching
    f. โปรเซสที่ออกแบบให้เป็น thread pool นั้นจะไม่มีการสร้างเธรดใหม่ขึ้นมาเลย มีแต่การเปลี่ยนสถานะเธรดระหว่าง ทำงาน และ พักคอย

12. ข้อความเปรียบเทียบระหว่าง preemptive และ non-preemptive scheduling ใดไม่ถูก
    a. Preemptive scheduling มีการเปลี่ยนสถานะของโปรเซสหลายรูปแบบกว่า
    b. Non-preemptive scheduling ไม่จำเป็นต้องใช้ความสามารถพิเศษของฮาร์ดแวร์เช่นเดียวกับที่ preemptive scheduling ต้องการ
    c. ถ้าทุกโปรเซสเป็น I/O Bound, preemptive scheduling ไม่ได้ช่วยให้การตอบสนองของระบบดีขึ้น เนื่องจากจะมีคอขวดที่อุปกรณ์ไอโอ
    d. Preemptive scheduling รองรับ scheduling แบบ round robin แต่ non-preemptive ไม่รองรับ
    e. ทั้งสองรูปแบบจะมีการทำงานของ scheduler ใน kernel mode
    f. การทำ kernel-level preemption นั้นซับซ้อนกว่าการทำ user-level preemption เพราะว่าต้องเก็บสถานะของโครงสร้างข้อมูลต่างๆ ของ kernel ไว้

13. ข้อความใดต่อไปนี้ที่เกี่ยวกับ dead lock และ starvation ผิด

I. ทั้ง deadlock และ starvation จะเกิดขึ้นได้ จะต้องมีโปรเซสมากกว่าหนึ่งโปรเซสในระบบ

II. หากระบบปฏิบัติการบังคับไม่ให้โปรเซสทำการขอทรัพยากรแบบ hold & wait ได้ แต่บังคับไม่ให้ขอทรัพยากรแบบ mutual exclusion ไม่ได้ deadlock ก็สามารถเกิดได้อยู่ดี

III. การใช้ semaphore สามารถทำให้เกิด deadlock ได้

IV. ในกรณีที่โปรเซสใช้ทรัพยากรร่วมกัน แต่ไม่มีการป้องกัน critical section ในการขอใช้ทรัพยากรนั้นๆ เลย จะไม่ทำให้เกิด deadlock

a. I, III.

b. II, IV.

c. IV.

d. II, III.

e. II, III, IV.

f. II.

รหัสนักศึกษา _____

ส่วนที่ 2 Find out about 'Find out more' (2 pts for each problem)

1. ในระบบปฏิบัติการ Windows NT family เราจะดูว่าอุปกรณ์ไอโอตัวไหนมีการทำ memory-mapped address ได้อย่างไร และอะไรคือข้อดีในการทำ memory-mapped ของอุปกรณ์เหล่านี้ (2 คะแนน)

2. อะไรคือ binary semaphore และเมื่อไหร่ควรจะใช้ binary semaphore (2 คะแนน)

รหัสนักศึกษา _____

3. จาก Bakery algorithm

```
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: }
```

1) ตัวแปรใดบ้างที่เป็น shared variable (1 คะแนน)

2) ถ้าเราตัด บรรทัดที่ 2, 4 และ 6 ทิ้ง algorithm นี้จะผิด จงอธิบายว่ามันจะผิดอย่างไร (2 คะแนน)

รหัสนักศึกษา _____

ส่วนที่ 3 ตอบคำถามในที่ที่กำหนดให้ (พยายามตอบให้สั้นได้ใจความ)

1. สมมุติว่า P1 ทำงานอยู่ (running state) และมี timer interrupt เข้ามา ระบบปฏิบัติการพบว่า time slice ของ P1 หมดแล้วจึงเลือก P2 ซึ่งพร้อมจะทำงาน (ready) มาทำงานแทน จงเติมขั้นตอนของสิ่งต่างๆ ที่จะเกิดขึ้น พร้อมระบุว่า ขณะนั้นโปรเซสเซอร์อยู่ในโหมดใด (2 คะแนน)

ขั้นตอน	โหมดการทำงาน
1. P1 ทำงาน มี timer interrupt เข้ามา	user
2. _____	kernel
3. ตรวจสอบว่า P1 ทำงานครบตาม time slice แล้ว	_____
4. _____	_____
5. _____เลือก P2 มาทำงานแทน (ส่วนใดของ OS เลือก P2)	kernel
6. _____	kernel
7. P2 ทำงาน	_____

2. Code นี้เป็น code ของ test_and_set ที่ทำงานได้ถูกต้อง

```
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:        /* critical section */
7:      j = (i+1) % n;
8:      while (( j != i) && !waiting[j])
9:        j = (j + 1) % n;
10:     if (j == i)
11:       lock = false;
12:     else
13:       waiting[j] = false;
14:     /* remainder section */
15: }
```

1) ถ้าเปลี่ยน บรรทัดที่ 9 เป็น

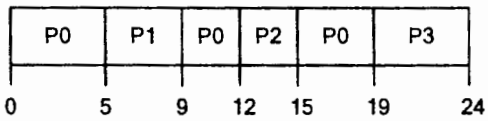
```
9:        j = (j - 1) % n
```

การทำงานจะเปลี่ยนไปเช่นไร และยังทำงานได้ถูกหรือไม่ (1 คะแนน)

รหัสนักศึกษา _____

2) ตัวแปรใดบ้างเป็น shared variable (1 คะแนน)

3. ถ้า Gantt chart มีลักษณะดังนี้



และข้อมูลของโปรเซสคือ

Process	Arrival time	Burst	Priority
P0	0	12	2
P1	5	4	1
P2	12	3	4
P3	14	5	0

ค่า priority ต่ำหมายถึงโปรเซสมี priority ต่ำ (เช่นค่า 0 มี priority ต่ำกว่า 1)

1) ระบบนี้ ใช้ scheduling algorithm ใด (จากที่เรียนมา ไม่รวมพวก multilevel) (1 คะแนน)

2) หาค่า waiting time ของแต่ละ process และค่าเฉลี่ย waiting time? (1 คะแนน)