ที่นั่งสอบ _____

King Mongkut's University of Technology Thonburi

Midterm Exam of Second Semester, Academic Year 2008

COURSE CPE ~~334~~ 330 Operating Systems Computer Engineering Department, 3rd Yr.

Monday 22 December 2008 ภาคภาค 2/2551. 9.00-12.00

Instructions

1. This examination contains 2 parts, 21 problems, 13 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

ฉลาดชาติ .

รหัสนักศึกษา _____

CPE 330 Operating Systems - Midterm Examination

ชื่อ _____

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. ชื่อใดต่อไปนี้ไม่ใช่และไม่เกี่ยวข้องกับระบบปฏิบัติการ
   a. Ubuntu
   b. Vista
   c. OS X
   d. Solaris
   e. ARM
   f. Symbian

2. ข้อใดไม่ใช้หน้าที่หรือความรับผิดชอบหลักของระบบปฏิบัติการทั่วไป
   a. สร้างโปรเซสใหม่
   b. หยุดการทำงานของโปรเซสชั่วคราว
   c. ลดภาระการทำงานของ CPU โดยใช้ screen saver
   d. จัดการระบบไฟล์และดิสก์
   e. ตรวจสอบการ log in เข้าระบบของผู้ใช้
   f. ตรวจสอบว่ามีอุปกรณ์มาต่อเพิ่มเติมกับระบบหรือไม่

3. ข้อความใดที่เกี่ยวกับ user/kernel mode ไม่ถูกต้อง
   a. ในการทำงานใน user mode โปรเซสไม่สามารถประมวลผลคำสั่งไอโอได้
   b. ใน user mode ระบบจะทำงานได้เร็วกว่าใน kernel mode เพราะไม่มีการ context switch
   c. Admin. ของระบบ จะสามารถเรียกโปรเซสให้ทำงานในโหมด kernel ไม่ได้
   d. เมื่อเกิด interrupt ขึ้น ไม่จำเป็นว่าจะต้องมีการเปลี่ยนโหมดเสมอไป แต่จะทำให้งานปัจจุบันหยุด (ชั่วคราว) เสมอ
   e. System call จะถูกเรียกเมื่อโปรเซสอยู่ใน user mode เท่านั้น
   f. หากโปรเซสเซอร์ไม่มี mode bit ระบบปฏิบัติการจะไม่สามารถใช้คุณสมบัติ user/kernel mode ได้

4. ข้อความใดที่เกี่ยวกับโครงสร้างของ kernel ผิด
   I. Kernel ตัวเดียวสามารถมีโครงสร้างได้ทั้ง monolithic และ microkernel
   II. Monolithic kernel จะมีขนาดใหญ่และเร็วกว่า microkernel เสมอ
   III. NT executive เป็นส่วนที่ไม่จำเป็นต้องอยู่ใน kernel mode ก็ได้ แต่ถูกจัดให้อยู่ใน kernel mode เพื่อเพิ่มประสิทธิภาพ

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

5. เรียงลำดับขั้นตอนที่ระบบปฏิบัติการและส่วน hardware จะต้องทำเมื่อมี hardware interrupt เกิดขึ้น

   I. Disable interrupt อื่นที่อาจจะเข้ามา

   II. หยุดการทำงานของโปรเซส(ที่ทำงานอยู่)ชั่วคราว

   III. ส่งสัญญาณตอบรับไปยังตัวควบคุม interrupt

   IV. ทำงานที่ interrupt service routine

   V. เก็บค่าของ Register ต่างๆ รวมทั้งค่า IRQ

   VI. ดูค่าในตาราง interrupt vector

   a. II, V, VI, III, IV.

   b. III, I, V, VI, IV, II.

   c. I, III, II, IV, V.

   d. VI, II, III, IV, I, V.

   e. V, II, I, III, IV, VI.

   f. II, IV, V, I, III.

6. ข้อความใดที่เกี่ยวกับ system call และพารามิเตอร์ของ system call ถูก

   a. การเรียก system call จะทำให้โปรเซสเปลี่ยนสถานะของการทำงาน (running, ready, waiting) เสมอ

   b. ระบบปฏิบัติการสามารถส่งค่าพารามิเตอร์ผ่านรีจิสเตอร์ได้เมื่อชนิดของพารามิเตอร์เป็น integer เท่านั้น เนื่องจากขนาดของรีจิสเตอร์ไม่ใหญ่พอที่จะเก็บค่า String

   c. การส่งค่าพารามิเตอร์ผ่าน stack เป็นวิธีที่เหมาะสมกับพารามิเตอร์ที่มีขนาดสั้นๆเท่านั้น

   d. การเตรียมการสำหรับส่งค่าพารามิเตอร์ของ system call ปรกติจะทำโดยฟังก์ชั่น ใน system library ซึ่งทำงานใน user mode

   e. ระบบปฏิบัติการสามารถเลือกใช้รีจิสเตอร์ตัวไหนเก็บค่าพารามิเตอร์ของ system call ก็ได้ เพราะค่าในรีจิสเตอร์จะถูกเก็บใน stack ก่อนแล้ว

   f. ในการส่งค่าพารามิเตอร์ผ่าน stack โปรเซสจะส่งค่าพารามิเตอร์ไปที่ kernel-mode stack เพื่อให้ system call routine รับค่าได้โดยตรง

7. ข้อใดไม่ใช่ความแตกต่างระหว่างโปรแกรมกับโปรเซส
   a. โปรเซสมี process descriptor โปรแกรมไม่มี
   b. โปรเซสมี stack segment โปรแกรมไม่มี
   c. โปรเซสมี context โปรแกรมไม่มี
   d. โปรแกรมถูกเก็บเป็นไฟล์ โปรเซสอยู่ในหน่วยความจำ
   e. โปรแกรมเก็บ code ในรูปแบบภาษาเครื่อง โปรเซสเก็บใน code segment
   f. โปรแกรมไม่มีการเก็บ heap แต่โปรเซสมี

8. ข้อใดกล่าวถึงสาเหตุและสถานะได้ถูกต้อง เมื่อโปรเซส P1 ต้องเปลี่ยนสถานะจาก running
   I. P1 จะถูกเปลี่ยนเป็น waiting หาก P1 ยอมปล่อย CPU (yield)
   II. P1 จะถูกเปลี่ยนเป็น ready หากมี P2 ที่มี priority ต่ำกว่า P1 เข้ามา และ scheduling algorithm เป็น priority scheduling ชนิด preemptive
   III. P1 จะถูกเปลี่ยนเป็น waiting หากมี P2 ที่มี priority สูงกว่า P1 เข้ามา และ scheduling algorithm เป็น priority scheduling ชนิด non-preemptive
   IV. P1 สามารถกลับมาเป็นสถานะ ready ได้ทันทีที่ time slice หมด และ scheduling algorithm เป็น round robin หากไม่มีโปรเซสอื่นคอยโปรเซสเซอร์อยู่

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

9. ข้อความใดที่เกี่ยวกับ polling และ interrupt ผิด
   a. ถ้าอุปกรณ์ไอโอใช้เวลาสั้นมากในการทำงานที่ได้รับมอบหมาย การทำ interrupt จะดีกว่า polling เพราะการ overhead ของ polling จะสูงกว่า interrupt
   b. DMA จะใช้การ interrupt เป็นการบอกให้ kernel รู้ว่าได้ทำงานเสร็จเรียบร้อยแล้ว
   c. ในรูปแบบ interrupt เมื่ออุปกรณ์ไอโอทำงานเสร็จจะแจ้งให้ CPU ทราบผ่าน device controller โดยมีการ set ค่า IRQ
   d. ในรูปแบบ interrupt จะสามารถทำให้เกิด context switch ได้ เมื่อมี IRQ เข้ามา
   e. การยอมให้มี nested interrupt ทำให้ระบบมีการตอบสนองต่ออุปกรณ์ไอโอที่ดีกว่าแต่จะมีความยุ่งยากเพิ่มขึ้นเนื่องจากสามารถมี interrupt service routine มากกว่าหนึ่ง routine ทำงานอยู่ใน kernel mode
   f. จะมีการตรวจสอบว่ามี interrupt เข้ามาที่โปรเซสเซอร์หรือไม่ ก่อนที่จะ execute ทุกๆ คำสั่ง (ภาษาเครื่อง)

10. ข้อความใดที่เกี่ยวกับการ boot ระบบ (สำหรับเครื่อง PC ทั่วไป) ผิด
   a. ส่วนหนึ่งของ BIOS จะเก็บค่าพารามิเตอร์ของ hard disk เพื่อช่วยในการหาตำแหน่งของ boot sector
   b. จะต้องมีการตรวจสอบอุปกรณ์สำคัญของระบบ ที่เรียกว่า POST เสมอ หากมีอาการผิดปกติ จะแสดงโดยการ beep
   c. ในการทำ dual boot จะต้องกำหนดค่าการ boot ใน boot sector ว่าจะให้ boot จาก partition ใดบ้าง
   d. Code ใน boot sector นั้นไม่สามารถจะโหลดตัว image ของ OS ทั่วไปได้ เนื่องจาก boot sector มีขนาดเล็กทำให้ไม่สามารถบรรจุรหัสได้เพียงพอ
   e. ข้อมูลใน BIOS จะถูกเก็บในหน่วยความจำประเภทไม่สูญหายเมื่อดับไฟ เช่น flash memory
   f. ไม่มีข้อความใดผิดเลย หรือ มีข้อความที่ผิดมากกว่าหนึ่งข้อ

รหัสนักศึกษา _____

11. ข้อความที่เกี่ยวกับสถานะของโปรเซสต่อไปนี้ข้อความใดบ้างที่ถูก (กำหนดให้เป็น single core)

I. จะมีเพียงโปรเซสเดียวเท่านั้นที่จะอยู่ในสถานะ running

II. จะต้องมีโปรเซสอยู่ใน ready queue เสมอ

III. ในสถานะ terminated โปรเซสจะต้องเป็น zombie ก่อนเสมอแล้วจึงจะถูกล้างจากระบบ

IV. หากโปรเซสเปลี่ยนสถานะจาก waiting ไปเป็น ready อาจมีการเรียก scheduler ได้ในบางกรณี

V. ใน non-preemptive scheduling จะไม่มีการเปลี่ยนสถานะจาก running ไปเป็น ready โดยตรง

a. I, II, V.

b. II, III, IV.

c. I.

d. I, IV, V.

e. I, IV.

f. I, III, IV,

12. ข้อความใดที่เกี่ยวกับ process communication ผิด

a. ถ้าเป็น blocking send โปรเซสที่ส่งข้อความจะต้องรอให้ผู้รับมารับเสมอ ไม่สามารถทำอย่างอื่นต่อเลยได้ถ้าผู้รับยังไม่รับข้อความ

b. การสร้าง pipe ที่ถูกต้องนั้น pipe จะถูกสร้างทันทีหลังจากการเรียก fork() system call

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

d. ตัว IPC semaphore อาจจะยังไม่ถูกลบแม้ว่าโปรเซสที่ใช้ semaphore จบการทำงานและออกจากระบบแล้ว

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

f. ไม่มีข้อความใดผิดเลย หรือ มีข้อความที่ผิดมากกว่าหนึ่งข้อ

13. อะไรเป็นสิ่งที่ thread ต่างๆ ต้อง share ร่วมกับ thread อื่นๆ ในโปรเซสเดียวกัน

a. Stack (user level)
b. Thread control block
c. Thread context
d. Register values
e. Execution code
f. ผิดทุกข้อ หรือมีมากกว่าหนึ่งสิ่ง

14. ข้อความใดที่เกี่ยวกับ thread ถูก

a. การสร้างเธรดทำได้เร็วกว่า การสร้างโปรเซสใหม่ เนื่องจากเธรดใช้หน่วยความจำน้อยกว่าโปรเซส
b. ใน M:1 thread model เธรดของโปรเซส P1 สามารถ block การทำงานของ เธรดของโปรเซส P2 หรือโปรเซสอื่นๆ ได้เนื่องจากทั้งสองเธรดจะ share kernel-level เธรดเดียวกัน
c. เธรดแบบ M:N จะมีความสามารถในการรองรับจำนวนเธรดได้มากกว่าใน model อื่น เนื่องจากจำนวนเธรดจะเท่ากับ M x N
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 มีแต่การเปลี่ยนสถานะของเธรดระหว่าง ทำงาน และ พักคอย

15. ข้อความเปรียบเทียบระหว่าง preemptive และ non-preemptive scheduling ใดไม่ถูก

a. Preemptive scheduling มีการเปลี่ยนสถานะของโปรเซสหลายรูปแบบกว่า
b. Non-preemptive scheduling ไม่จำเป็นต้องใช้ความสามารถพิเศษของฮาร์ดแวร์เช่นเดียวกับที่ preemptive scheduling ต้องการ
c. ถ้าโปรเซสส่วนมากเป็น I/O Bound, preemptive scheduling จะช่วยให้การตอบสนองของระบบดีขึ้นเนื่องจากจะทำให้โปรเซสคอยใน wait queue แทนการคอย I/O ทำงานเสร็จใน kernel
d. Preemptive scheduling รองรับ scheduling แบบ round robin แต่ non-preemptive ไม่รองรับ
e. Non-preemptive scheduling จะทำให้ระบบมีประสิทธิภาพต่ำกว่า preemptive เสมอ
f. การทำ kernel-level preemption นั้นซับซ้อนกว่าการทำ user-level preemption เพราะว่าต้องเก็บสถานะของโครงสร้างข้อมูลต่างๆ ของ kernel ไว้

16. ข้อความใดต่อไปนี้ที่เกี่ยวกับ dead lock และ starvation ผิด

I. ทั้ง deadlock และ starvation จะเกิดขึ้นได้ จะต้องมีโปรเซสมากกว่าหนึ่งโปรเซสในระบบ

II. การใช้ semaphore เพียงตัวเดียว (และเป็นทรัพยากรที่ใช้ร่วมกันชนิดเดียว) ก็สามารถทำให้เกิด deadlock ได้

III. หากระบบปฏิบัติการบังคับไม่ให้โปรเซสทำการขอทรัพยากรแบบ circular wait ได้ แต่บังคับไม่ให้ขอทรัพยากรแบบ mutual exclusion ไม่ได้ deadlock ก็สามารถเกิดได้อยู่ดี

IV. ระบบปฏิบัติการที่มีการใช้ dynamic priority จะช่วยลดปัญหาการเกิด starvation ขึ้น

a. I, II.

b. III, IV.

c. IV.

d. II, III.

e. II, III, IV.

f. III.

รหัสนักศึกษา _____

ส่วนที่ 2 ตอบคำถาม

1. จาก find out more ใน lecture และจาก minithreads

   a. ข้อดีของ monitor เมื่อเทียบกับ semaphore คืออะไร ยกตัวอย่างมาสักหนึ่งอัน (1 คะแนน)

   b. อะไรคือความแตกต่างระหว่าง food service กับ producer consumer (1 คะแนน)

2. จงยกตัวอย่าง race condition ที่เกิดขึ้นใน ปัญหา readers writers โดยเลือกสองโปรเซสใดๆ (ให้บอกด้วยว่าเป็น reader หรือ writer) และ แสดงการ context switch ในรูปแบบที่จะทำให้เกิด race condition สามารถใช้เลขบรรทัดแทนได้ (2 คะแนน)

```
    Reader                                      Writer
R1  wait(mutex);                          W1    wait(wrt);
R2        readcount++;                    W2          writing(); // CS
R3        if (readcount == 1)wait(wrt);   W3    signal(wrt);
R4  signal(mutex);
R5        reading(); // CS
R6  wait(mutex);
R7        readcount--;
R8        if (readcount == 0)signal(wrt);
R9  signal(mutex);
```

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: }
```

a. ตัวแปร number[] จำเป็นต้องเป็น shared variable เพราะอะไร (1 คะแนน)

b. ถ้าเราตัด บรรทัดที่ 6 ทิ้ง algorithm นี้จะผิด จงยกตัวอย่างว่ามันจะผิดอย่างไร (1 คะแนน)

4. จากเนื้อหาที่บอกว่า การทำงานภายใน wait(), signal() ของ semaphore จะต้องเป็น atomic

```
wait(s) {
        while (s<= 0) do no-op;
        s--;
}
```

```
signal(s) {
        s++;
}
```

หากทำให้ wait() เป็น atomic แต่เมื่อการทำงานเข้าไปอยู่ใน no-op ก็จะหมายความว่า ไม่มีโปรเซสอื่นใดสามารถเรียก signal() ได้ และจะหมายถึงว่า wait() จะต้องคอยตลอดไปเพราะค่า s ไม่มีทางเปลี่ยน ให้อธิบายว่าความหมายที่แท้จริงของ 'จะต้องเป็น atomic' คืออะไร พร้อมยกตัวอย่าง (2 คะแนน)

รหัสนักศึกษา _____

5. กำหนดให้ time slice = 20 units, context switch = 1 unit และมี process burst ดังนี้

Process	Arrival time	Burst
P0	0	42
P1	5	8
P2	23	60
P3	49	16

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

b. วาด Gantt chart ของการ schedule โปรเซสเมื่อใช้ shortest remaining time first algorithm พร้อมหาค่า average waiting time (1 คะแนน)