



**King Mongkut's University of Technology Thonburi**

**Final Exam of Second Semester, Academic Year 2007**

**COURSE** CPE 330 Operating Systems      **Computer Engineering Department, 3<sup>rd</sup> Yr.**

**Wednesday 5 March 2008**

**9.00-12.00**

**Instructions**

1. This examination contains 3 parts, 19 problems, 11 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**

## CPE 330 Operating Systems - Final Examination

ชื่อ \_\_\_\_\_

**ส่วนที่ 1** เลือกข้อที่ถูกที่สุด (12 คะแนน ข้อละคะแนน) ในแต่ละคำถามจะมีคำตอบที่ถูกอยู่เพียงหนึ่งคำตอบ คุณสามารถเลือกทำสิ่งต่างๆ เหล่านี้ได้ในแต่ละข้อ

1. ปลอยให้ว่าง – 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. ข้อใดไม่ถูกต้องในการกล่าวถึงตำแหน่งอ้างอิง (address) และการเชื่อมตำแหน่ง (binding)
  - a. ระหว่างเขียนโปรแกรม (source code) จะยังไม่มี การเชื่อมกับตำแหน่งใดๆ ในหน่วยความจำ การอ้างอิงต่างๆ จะถูกทำผ่านชื่อของ ค่าคงที่ ตัวแปร และ ฟังก์ชัน
  - b. หลังจากคอมไพล์โปรแกรมเป็น object code จะยังไม่มี การเชื่อมกับตำแหน่งใดๆ ผ่านเคอร์เนลเลย การอ้างอิงต่างๆ จะถูกทำผ่านตำแหน่งแบบ relocatable ภายใน object ไฟล์เท่านั้น
  - c. หลังจากการลิงค์โปรแกรมกับ library ต่างๆ แล้ว ตำแหน่งอ้างอิงบางส่วนยังสามารถอยู่ในรูป relocatable ได้
  - d. โปรแกรมสองโปรแกรม (คนละโปรแกรม) สามารถมีส่วน code ที่มีช่วงของตำแหน่ง relocatable ซ้อนทับกันได้
  - e. โปรแกรมสองโปรแกรมที่เกิดจากโปรแกรมเดียวกัน สามารถมีส่วน data ที่มีช่วงของตำแหน่ง relative ซ้อนทับกันได้
  - f. เคอร์เนลจัดการแปลงตำแหน่ง relocatable ไปสู่ตำแหน่ง logical ในขณะที่ลิงค์โปรแกรม

2. จากข้อความต่อไปนี้ซึ่งเกี่ยวข้องกับการลิงค์ประเภทต่างๆ จงระบุว่าข้อความใดบ้างที่ไม่ถูก
  - I. สำหรับโปรแกรมเดียวกัน binary image ของการลิงค์แบบ static มักจะใหญ่กว่า แบบ dynamic เพราะต้องดึงรหัสของ library มารวมด้วย
  - II. Dynamic linking ใช้สำหรับกรณีที่โปรแกรมประกอบด้วยหลายๆ object ไฟล์เท่านั้น เพราะถ้ามีอยู่เพียงไฟล์เดียว จะสามารถรู้ตำแหน่งต่างๆ ภายในได้อยู่แล้ว
  - III. ในโปรแกรมหนึ่งอาจจะสามารถมีการลิงค์แบบ static และ dynamic loading พร้อมๆ กันได้
  - a. I.
  - b. II.
  - c. III.
  - d. I, II.
  - e. II, III.
  - f. I, III.
  
3. จากข้อความต่อไปนี้ ซึ่งเกี่ยวข้องกับชนิดของไฟล์ library ที่ใช้ในการลิงค์ จงระบุว่าข้อความใดบ้างที่ถูก
  - I. DLL ไฟล์จะถูกใช้สำหรับการลิงค์ทั้ง static และ dynamic linking
  - II. SO ไฟล์จะมีลักษณะคล้าย LIB ไฟล์คือเป็น Static Object ไม่ถูกในหลังโปรเซสทำงาน
  - III. LIB ไฟล์จะถูกใช้ในขั้นตอน linking เท่านั้น ไม่จำเป็นต้องมีอยู่ในระบบ เมื่อโปรเซสทำงาน
  - a. II, III.
  - b. ถูกทุกข้อความ
  - c. I.
  - d. III.
  - e. ไม่มีข้อความใดถูกเลย
  - f. I, III.
  
4. ข้อความใดที่เกี่ยวกับ fragmentation ในหน่วยความจำ ผิด
  - a. Hole สามารถเกิดได้จาก fragmentation ทั้งสองแบบ
  - b. ระบบปฏิบัติการต้องการลดหรือกำจัด fragmentation ในหน่วยความจำ หากมันสามารถทำได้
  - c. Internal fragmentation ไม่ส่งผลกระทบต่อประสิทธิภาพการทำงานของระบบโดยตรง
  - d. ระบบปฏิบัติการจะต้องคอยจัดจำเนื้อที่ว่างซึ่งเกิดจาก external fragmentation
  - e. ในการออกแบบการจัดการหน่วยความจำ อาจเกิด fragmentation ทั้งสองแบบในเวลาเดียวกัน
  - f. เมื่อเวลาผ่านไป จำนวน hole และขนาดของมัน สามารถเพิ่มขึ้น และ ลดลง ได้

5. จากข้อความต่อไปนี้ซึ่งเกี่ยวข้องกับ paging และ segmentation ข้อความใดบ้างที่ถูก
- I. เพจต่างๆ เพจในระบบใดระบบหนึ่งจะมีขนาดเท่ากันและเท่ากับขนาดของเฟรมด้วย
  - II. เซกเมนต์แต่ละเซกเมนต์สามารถมีขนาดแตกต่างกันได้และอาจแตกต่างจากขนาดของเพจ
  - III. โดยปกติวงจรใช้ที่จัดการกับการแปลงเพจและเซกเมนต์ (แปลงรูปแบบของ memory addressing) จะอยู่ภายในเคอร์เนล
  - IV. ในการทำ paging จะไม่ทำให้เกิด fragmentation เพราะไม่เกิด hole
- a. I, II, IV.
  - b. II, IV.
  - c. I, II.
  - d. I, III, IV.
  - e. I, II, III, IV.
  - f. II, III, IV.
6. ข้อความใดที่เกี่ยวกับ virtual memory ไม่ถูก
- a. ขนาดของโปรเซสและเคอร์เนล รวมกันทั้งหมดสามารถใหญ่เกินกว่าขนาด RAM ได้ แต่มันต้องมีขนาดเล็กกว่า พื้นที่ในฮาร์ดดิสก์ซึ่งถูกจองไว้สำหรับการสลับเพจ
  - b. ใน page table จะต้องมีการเก็บสถานะ valid/invalid หากมีการรองรับ virtual memory
  - c. การ swapping ไม่ถูกว่า เป็นการทำให้ virtual memory ถึงแม้ว่าจะมีการย้ายตัวโปรเซสระหว่าง RAM และ ฮาร์ดดิสก์ก็ตาม
  - d. หากพื้นที่ว่างใน RAM ใหญ่กว่าขนาดของโปรเซสตลอดเวลา page fault ก็ยังสามารถเกิดขึ้นได้กับโปรเซสนั้น
  - e. เมื่อเกิด page fault บางครั้งอาจจะไม่จำเป็นที่จะต้องหา victim (เพจที่ถูกสลับลงไปฮาร์ดดิสก์)
  - f. มีข้อความไม่ถูกมากกว่าหนึ่งข้อ หรือ ทุกข้อความข้างบนถูกทั้งหมด
7. อะไรต่อไปนี้ไม่เป็นส่วนหนึ่งในการจัดการ page fault ไม่ว่ากรณีใดๆ
- a. มีการปรับค่าในตารางเพจ
  - b. มีการเปลี่ยน mode ระหว่าง user และ kernel mode
  - c. มีการเขียนข้อมูลลงดิสก์
  - d. มีการทำงานของ page replacement
  - e. มีการอ่านข้อมูลจากดิสก์
  - f. ทุกส่วนเกี่ยวข้องกับการจัดการ page fault ทั้งหมด

8. หากเวลาเฉลี่ยในการเข้าถึง RAM = 10 ns เวลาเฉลี่ยในการทำ page fault service = 20 ms และเรา ยอมรับเวลาเฉลี่ยในการเข้าถึงจริง (effective access time) อย่างมาก 15 ns ค่า page fault rate ควร จะมีค่าไม่เกินค่าใด (ค่าโดยประมาณ)
- $1 \times 10^8$
  - $1 \times 10^6$
  - $2 \times 10^6$
  - $2.5 \times 10^7$
  - $3 \times 10^7$
  - ยากเกินกว่าที่จะคิดได้
9. จากข้อความต่อไปนี้ซึ่งเกี่ยวข้องกับ page replacement algorithm ข้อความใดบ้างที่ถูก
- FIFO และ LRU สามารถทำให้เกิด belady's anomaly
  - LRU และ Optimal สามารถถูกพัฒนาให้ใช้ได้จริงในระบบปฏิบัติการ
  - การใช้ Approximate LRU จะมี overhead น้อยกว่าการใช้ LRU
  - LRU-K เป็นการประมาณ LRU แบบหนึ่งซึ่งจะใช้ K additional bits ในการหา victim
- I, III.
  - III, IV.
  - I, II, III.
  - II, IV.
  - III.
  - IV.
10. จงจับคู่ระหว่าง allocation method และ free space management ซึ่งมีการจัดการแบบเดียวกัน
- Allocation methods: Contiguous (con), Link allocation (lnk), index allocation (idx)
  - Free space managements: Linked list (L), Grouping (G), Counting (C)
- (con:L) , (lnk:G) , (idx, C)
  - (con:L) , (lnk:C) , (idx, G)
  - (con:G) , (lnk:L) , (idx, C)
  - (con:G) , (lnk:C) , (idx, L)
  - (con:C) , (lnk: L) , (idx, G)
  - (con:C) , (lnk:G) , (idx, L)

11. ข้อความใดที่เกี่ยวข้องกับฮาร์ดดิสก์เทคโนโลยี ถูก
  - a. หน่วยย่อยๆ ที่สุดในการจัดเก็บข้อมูลคือ sector ซึ่งโดยปกติจะมีขนาดเป็นค่ากำลังสองใดๆ เช่น 1KB, 4KB
  - b. เวลาส่วนมากในการย้ายหัวอ่านไปยังพื้นผิวที่เก็บข้อมูลที่ต้องการ คือ rotational latency เนื่องจากฮาร์ดดิสก์หมุนช้า (ไม่ถึงหมื่นรอบต่อวินาที) ในขณะที่ CPU ทำงานได้เป็น พันล้านคำสั่งต่อวินาที
  - c. ฮาร์ดดิสก์แบบ SATA-II จะมีความเร็วสูงสุดในการเขียนข้อมูลลงจาน ที่ 3Gbits ต่อวินาที
  - d. ใน RAID-1 หากฮาร์ดดิสก์สองตัวในระบบมีความจุรวม 1 TB จะเก็บข้อมูลจริงได้ 500 GB
  - e. Disk buffer และ track buffer อยู่ในอุปกรณ์ฮาร์ดดิสก์เหมือนกัน และถูกใช้งานลักษณะเดียวกัน
  - f. Command queuing เป็นการจัดคิวของการร้องขอ block ของฮาร์ดดิสก์ซึ่งจัดการโดยระบบปฏิบัติการเพื่อเพิ่มประสิทธิภาพโดยรวมของระบบ
  
12. ข้อความใดเกี่ยวข้องกับ disk scheduler algorithm ถูก
  - a. Algorithm complete fair queue เป็น algorithm ที่เดียวที่ถูกใช้ในลินุกซ์รุ่นปัจจุบัน
  - b. Algorithm anticipatory จะมีการใช้รูปแบบการร้องขอ disk block ในอดีต สำหรับการคาดการณ์ตำแหน่งของ disk block ที่จะถูกร้องขอในอนาคต
  - c. Disk scheduler จะมีผลมากเมื่อระบบมีการร้องขอ disk block น้อย (light load) เนื่องจากการร้องขอแต่ละครั้งจะสามารถมีตำแหน่งที่ห่างกันมาก
  - d. Rotational speed จะถูกนำมาคำนวณใน disk scheduler algorithm ด้วย
  - e. Variance (ความแปรปรวน) ของเวลาในการคอยการตอบสนองการร้องขอ ของ C-SCAN จะดีกว่า SSTF (shortest seek time first)
  - f. อาจเกิด starvation ได้ใน anticipatory algorithm ได้
  
13. ข้อความใดเกี่ยวข้องกับ file system ถูก
  - a. Superblock จำเป็นต้องเป็น block แรกในดิสก์เสมอ
  - b. ข้อแตกต่างหลักระหว่าง inode และ index ธรรมดา คือ inode มีการเก็บข้อมูลเกี่ยวข้องกับ file นั้นๆ ในโครงสร้างข้อมูล inode ด้วย
  - c. เป็นไปได้ว่ายังมี block ว่างเหลือในดิสก์แต่ระบบปฏิบัติการอาจไม่สามารถสร้างไฟล์เพิ่มได้อีก เนื่องจาก inode เต็ม
  - d. เนื่องจากโครงสร้าง inode ใช้เนื้อที่ไม่ถึง 1 disk block จึงสามารถนำที่ส่วนที่เหลือมาเก็บข้อมูลของไฟล์ได้ในบล็อกนั้นเลย โดยการอ้างอิงผ่าน inode ยังคงอยู่ในรูปแบบเดิม
  - e. หากระบบรองรับ directory จะต้องมีการกำหนดโครงสร้าง inode สำหรับ directory ขึ้นต่างหาก
  - f. ข้อความทุกข้อความผิดหมด หรือมีข้อความข้างบนที่ถูกมากกว่าหนึ่งข้อ

**ส่วนที่ 2** Find out about 'Find out more' จากที่เรียน (2 คะแนนในแต่ละข้อ)

1. จากการใช้โปรแกรมเพื่อดูว่าไฟล์ .EXE มีข้อมูลอย่างไรบ้าง พบข้อมูลในส่วน header ที่น่าสนใจคือ  
Size of code: 00005000h    Size of Initialized data: 00003000h    Base of Code: 00001000h  
Base of Data: 00006000h    Address of Entry Point: 004046F0h    Image Base: 00400000h  
จงวาด memory map ของโปรเซสนี้ ในรูปแบบ virtual address พร้อมระบุตำแหน่ง code ที่จะถูก  
ประมวลผลเป็นคำสั่งแรก และตำแหน่งของ code, data segments

2. อะไรคือ journalized file system (1 คะแนน), มีข้อดีอย่างไร มีข้อเสียอย่างไร (1 คะแนน)

3. ใน Microsoft Windows XP (คิดว่า Vista ก็คงคล้ายๆ กัน)
- a. จะสามารถ enable / disable virtual memory ได้อย่างไร (1 คะแนน)

- b. Windows เก็บ swapped pages ไว้ที่ใด (1 คะแนน)



### ส่วนที่ 3 ตอบคำถามในที่ที่กำหนดให้

ข้อควรระวัง ไม่มีคะแนนค่าน้ำหนักให้ในการตอบไม่ตรงประเด็น ถ้าตอบไม่ได้จริงๆ ควรเว้นไว้

1. โปรเซส P1 ใช้เนื้อที่หน่วยความจำทั้งหมด 30 MB หากระบบปฏิบัติการจัดการ paging ด้วยรูปแบบ 2-level paging โดยที่มีขนาด directory table (ระดับบน) 9 bits, page table 9 bits และ ขนาดเพจและเฟรม 2KB กำหนดให้รายการแต่ละรายการในตาราง (ทั้ง directory และ page table) ใช้เนื้อที่ 4 bytes จงคำนวณหาสิ่งต่างๆ ดังนี้
  - a. จำนวนรายการในตาราง directory และ จำนวนตารางเพจ (0.5 คะแนน)
  - b. จำนวนรายการในตารางเพจและ directory ทุกตารางรวมกัน (0.5 คะแนน)
  - c. จำนวนเฟรมที่ต้องการจัดเก็บโปรเซสทั้งหมด ทั้งตัวโปรเซสเอง และส่วนที่ใช้ไปสำหรับตารางทั้งหลาย (1 คะแนน)
  - d. หาก P1 ใช้ physical address ต่อเนื่องกันตั้งแต่ 12MB-42MB, จงระบุช่วงของค่าเฟรมที่ถูกเก็บในตารางเพจ (1 คะแนน)

