ชื่อ..... รหัสนักศึกษา.....

PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Midterm Examination: 1st Semester | Academic Year: 2009

Date: 28 July 2009 | Time: 13:30 – 16:30

Subject:(240-306) 241-304 Computer Operating Systems | Room: R300

ทุจริตในการสอบ โทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

Instruction:

- Please write your name and student id on all pages. There are 10 pages.
- This examination has 10 questions. Please answer all questions. Your answer can be in Thai.
- There is one extra credit question on the last page which is an optional question.

NOTE that I can only grade what I can read. If I cannot read your name or your id, you will not get the score.

คำสั่ง

- กรุณาเขียนชื่อและรหัสนักศึกษาบนข้อสอบทุกหน้า ข้อสอบมีทั้งหมด 10 หน้า
- ข้อสอบมี 10 ข้อ กรุณาตอบทุกข้อ คุณสามารถตอบเป็นภาษาไทยได้
- ข้อสอบพิเศษในหน้าสุดท้ายนั้นคุณจะตอบหรือไม่ก็ได้

อาจารย์จะสามารถตรวจได้เฉพาะคำตอบที่อาจารย์อ่านออกเท่านั้น หากอาจารย์อ่านคำตอบคุณไม่ออก หรือ อ่านชื่อและรหัสนักศึกษาของคุณไม่ออก คุณจะไม่ได้คะแนน

ชื่อ..... รหัสนักศึกษา.....

Question 1: (25 points) Describe the actions taken to context switch between processes.
อธิบายกระบวนการที่จะเกิดขึ้นเพื่อทำ context switch ระหว่างโพรเซส

Question 2: (25 points) Provide two programming examples in which multithreading model provides better performance than a single-threaded solution. Please explain how your examples use the multithreading model. เสนอตัวอย่างสองตัวอย่างที่โปรแกรมที่ใช้หลักการแบบมัลลิเธรดดิงให้ประสิทธิภาพดีกว่าโปรแกรมที่ใช้หลักการแบบเธรดเดียว และอธิบายว่าตัวอย่างของคุณใช้ระบบมัลติเธรดอย่างไร

Question 3: (15 points) Fill in the blank เติมคำในช่องว่าง

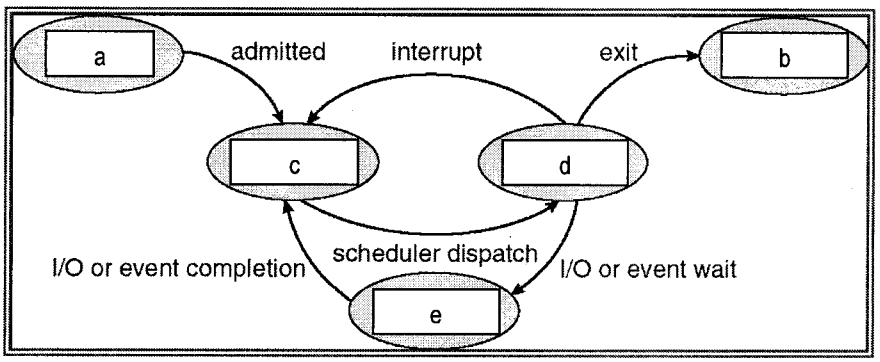

Diagram of process state

a. _____ b. _____

c. _____ d. _____ e. _____

ชื่อ..... รหัสนักศึกษา.....

Question 5: (50 points) Please select TRUE or FALSE (โปรดเลือกว่าข้อความต่อไปนี้ ถูก หรือ ผิด)

TRUE FALSE	When there are few other users, the task is large, and the hardware is fast, time-sharing system makes sense. ระบบ time-sharing เหมาะสมเมื่อระบบมีผู้ใช้ไม่มาก ลักษณะงานมีขนาดใหญ่ และตัวเครื่องมีความเร็วสูง
TRUE FALSE	In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in stealing or copying one's programs or data. ในสภาพแวดล้อมที่เป็นแบบ multiprogramming และ time-sharing นั้น ผู้ใช้งานหลายๆท่านจะใช้งานระบบร่วมกันในเวลาเดียวกัน ซึ่งลักษณะเช่นนี้สามารถทำให้เกิดการขโมยหรือคัดลอกโปรแกรมหรือข้อมูลของผู้อื่นได้
TRUE FALSE	We can ensure the same degree of security in a time-shared machine as in a dedicated machine. เราสามารถรับประกันระดับความปลอดภัยบนเครื่อง time-sharing ได้เหมือนเครื่องแบบ dedicated
TRUE FALSE	Virtual machines provide a good platform for operating system research since many different operating systems can run on one physical system. Virtual machines เป็นเวทีที่ดีสำหรับทำการวิจัยทางด้านระบบปฏิบัติการเนื่องเครื่องเดียวสามารถให้มีระบบปฏิบัติการหลายๆ ตัวรันอยู่ได้
TRUE FALSE	If the acquire() and release() semaphore operations are not executed atomically, then mutual exclusion may be violated. หากคำสั่ง acquire() และ คำสั่ง release() เซมะฟอร์ไม่ถูกเอ็กซะคิวแบบอะตอมมิก แล้วตัว mutual exclusion อาจถูกล่วงละเมิดได้
TRUE FALSE	FCFS scheduling algorithm discriminates against short jobs since any short jobs arriving after long jobs will have a longer waiting time. อัลกอริธึมการจัดลำดับงานแบบ FCFC นั้นไม่ให้ความยุติธรรมกับงานสั้นๆ เนื่องจากงานสั้นที่เข้ามาหลังงานขนาดยาวต้องรอนาน
TRUE FALSE	Remote Procedure Call (RPC) enables device controllers to inform the CPU that it has finished its operation. อาร์พีซี ทำให้ตัวควบคุมอุปกรณ์บอกกับหน่วยประมวลผลว่าตนเองนั้นทำงานเสร็จแล้ว
TRUE FALSE	A thread library provides the programmer with an API (Application Programming Interface) for creating and managing threads เธรดไลบรารีมีเอพีไอให้ผู้เขียนโปรแกรมใช้ในการสร้างและจัดการกับเธรด
TRUE FALSE	The main purposes of an operating system are (1) to allocate the resources and (2) to control the system หน้าที่หลักของระบบปฏิบัติการคือ การแจกจ่ายทรัพยากรของระบบและการควบคุมระบบ
TRUE FALSE	Interrupt allows user-level processes to request services of the operating system อินเทอร์รัพท์อนุญาตให้โพรเซสของผู้ใช้ร้องขอการบริการจากระบบปฏิบัติการ

ชื่อ..... รหัสนักศึกษา.....

b) (20 points) What is the turnaround time and waiting time of each process for each of these scheduling algorithms? (จงกรอกข้อมูลค่า waiting time ของโพรเซสแต่ละตัว เมื่อเราใช้สเก็ตดูลเลอร์แต่ละวิธี)

algorithm	Waiting time (milliseconds)				
	P1	P2	P3	P4	P5
First-come-First-Served					
Shortest-Job-First					
Non-preemptive priority					
Round-Robin					

c) (10 points) Which of the algorithms results in the minimum average waiting time (over all processes)? อัลกอรีทึมใดได้ค่าเฉลี่ยการรอคอย (average waiting time) ที่ต่ำที่สุด

Question 7: (25 points) What advantage is there in having different time-quantum sizes on different levels of a multilevel queuing system? จงอธิบายข้อดีของการมี time-quantum ที่แตกต่างกันของคิวแต่ละคิวในระบบแบบหลายคิว

Question 9: (25 points) Explain whether the following solution to the Reader-Writer problem can cause any problem and how?
จงอธิบายว่าโปรแกรมสำหรับแก้ปัญหาผู้อ่าน-ผู้เขียนที่ให้ข้างล่างนี้จะสร้างปัญหาใดได้บ้าง และ อย่างไร

```
int readerCount = 0;
Semaphore mutex = new Semaphore(1);
Semaphore db = new Semaphore(1);
```

```
/* reader */
while (true) {
  mutex.acquire();
  ++readerCount;
  if(readerCount == 1) { db.acquire(); }

// first reader
  mutex.release();

   // read from the database
   mutex.acquire();
  --readerCount;

  if(readerCount == 0) { db.release(); }

// last reader
  mutex.release();
}
```

```
/* writer */
while (true) {
  db.acquire();
  // update the database
  db.release();
}
```

```
/* semaphore acquire method */
acquire() {
 while value <= 0
   ; // no operation
 value--;
}

/* semaphore release method */
release(){
  value++;
}
```

ชื่อ..... รหัสนักศึกษา.....

ข้อมูลเพื่อใช้ในการตอบคำถาม

- Multiprogramming ระบบมัลติโปรแกรมมิ่งจะจัดสิ่งแวดล้อมให้ทรัพยากรต่างๆของระบบ เช่น ซีพียู หน่วยความจำ และอุปกรณ์เชื่อมต่ออื่นๆ ให้ถูกใช้ประโยชน์อย่างเต็มที่
- Time-sharing ระบบไทม์แชร์ริ่ง เป็นระบบที่ใช้ระบบมัลติโปรแกรมมิ่งเพื่อสลับเปลี่ยนการทำงานระหว่างงานหลายๆงานโดยสลับการทำงานอย่างรวดเร็วมากจนกระทั่งผู้ใช้ไม่สามารถสังเกตุเห็นได้ว่ามีงานอื่นๆ หรือมีผู้ใช้อื่นๆ ร่วมใช้งานระบบอยู่ในขณะเดียวกันด้วย
- Dedicated machine คือ การที่ใช้งานเครื่องโดยไม่แบ่งกับใครเลย
- Virtual machines คือ การปฏิบัติกับระบบปฏิบัติการเสมือนเป็นฮาร์ดแวร์ตัวหนึ่ง virtual machine จะให้ความรู้สึกเหมือนมีการเชื่อมต่อแบบเดียวกับฮาร์ดแวร์หลายๆ ชนิด
- Starvation ปัญหาการอดอยาก ซึ่งคือการที่โพรเซสหรืองานใดๆ ต้องรอเป็นระยะเวลาอันยาวนานหรือในบางกรณีเป็นการรอแบบไม่มีที่สิ้นสุด
- Preemptive เมื่อโพรเซสได้เริ่มรันแล้ว ระบบจะสามารถหยุดการทำงานของโพรเซสและเรียกทรัพยากรคืนเพื่อไปแจกจ่ายให้กับโพรเซสอื่นได้ ซึ่งในระหว่างนั้น ตัวโพรเซสที่โดนพรีเอ็มก็จะไปรออยู่ใน ready queue
- Non-preemptive เมื่อโพรเซสได้เริ่มรันแล้ว โพรเซสจะใช้งานซีพียูไปจนกระทั่งเสร็จสิ้นการทำงาน
- Waiting time เวลาที่โพรเซสต้องรออยู่ใน ready queue
- Mutual Exclusion: หมายถึงในเวลาขณะใดขณะหนึ่ง หากมีโพรเซสหนึ่งกำลังรันอยู่ใน critical section โพรเซสอื่นๆ จะไม่สามารถรันใน critical section
- Progress: หมายถึง หากไม่มีโพรเซสใดกำลังรันอยู่ใน critical section และมีโพรเซสอื่นๆ ที่กำลังรอจะเข้าไปรันใน critical section แล้วนั้น การเลือกโพรเซสเพื่อเข้าไปรันใน critical section จะไม่สามารถเลื่อนออกไปได้
- Bounded waiting: หมายถึง มีการกำหนดจำนวนครั้งหรือจำนวนโพรเซสอื่นซึ่งเข้าไปรันใน critical section ในระหว่างที่โพรเซสหนึ่งรอขอเข้าไปรันใน critical section จนถึงเวลาที่โพรเซสนั้นได้รับอนุญาตให้รันได้
- Atomic: อะตอมมิก คือ การกระทำงานซึ่งจะต้องทำให้เสร็จสิ้น หรือไม่ทำเลย มีแค่สองกรณีเท่านั้น

ชื่อ..... รหัสนักศึกษา.....

Question 4: (25 points) What would be the output from the following problem at LINE C and LINE P? Explain your answer
ผลลัพธ์ที่ LINE C และ LINE P คืออะไร โปรดอธิบายคำตอบของท่าน

```
#include <pthread.h>
#include <stdio.h>

int value = 15;
void *runner(void *param);

int main(int argc, char *argv[]) {
  pthread_t tid;
  pthread_attr_t attr;
  int pid;

 pid = fork();

 if (pid == 0) {
    pthread_attr_init(&attr);
    pthread_create(&tid,&attr,runner,NULL);
    pthread_join(tid,NULL);
    printf("CHILD: value = %d\n",value);  /* LINE C */
  }
  else if(pid > 0) {
    wait(NULL);
    printf("PARENT: value = %d\n",value);  /* LINE P */
  }
}

void *runner(void *param) {
  value = 20;
  pthread_exit(0);
}
```

ชื่อ..... รหัสนักศึกษา.....

Question 6: Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P1	7	2
P2	1	1
P3	2	2
P4	1	3
P5	5	4

The processes are assumed to have arrived in the order P1, P2, P3, P4, and P5. All processes arrived at time 0.
จากข้อมูลของโพรเซสทั้งห้าตัวที่กำหนดให้ข้างต้น สมมุติให้โพรเซสทั้งหมดเข้ามาในระบบตามลำดับ P1, P2, P3, P4, และ P5 ณ เวลา time = 0

a) (40 points) draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: First-Come-First-Served, Shortest-Job-First, Non-preemptive priority (a smaller priority number implies a higher priority use the arriving order to break tie), and Round-Robin (time quantum = 1). Use FCFS to break tie.
จงวาด Gantt charts เพื่อแสดงการทำงานของโพรเซสแต่ละตัว เมื่อเราใช้ First-Come-First-Served, Shortest-Job-First, Non-preemptive priority (ตัวเลขน้อยหมายถึงค่าไพออริทีมาก ถ้าไพออริทีเท่ากันเอาตัวที่มาก่อน) และ Round-Robin (time quantum = 2) หากมีตัวเลือกสองตัวให้เลือกทำงานโพรเซสที่เข้ามาก่อนเสมอ

First-Come-First-Served

Shortest-Job-First (preemptive)

Non-preemptive priority (a smaller priority number implies a higher priority) ตัวเลขน้อยหมายถึงค่าไพออริทีมาก

Round-Robin (time quantum = 2)

ชื่อ..... รหัสนักศึกษา.....

Question 8: (25 points) Following is the structure of process Pi in a two process synchronization problem where variable turn and flag are shared between the two processes, please explain whether the solution preserves the mutual exclusion and satisfies the progress and the bounded waiting requirements.

ข้างล่างคือโครงร่างโปรแกรมของโพรเซส Pi ในปัญหาการจัดการทำงานให้สอดคล้องกันของสองโพรเซส ทั้งสองโพรเซสจะมีการใช้ตัวแปร turn และตัวแปร flag ร่วมกันอยู่โปรดอธิบายว่าวิธีการดังกล่าวทำให้ mutual execution คงอยู่และสามารถรองรับ progress และ bounded waiting หรือไม่

```
int turn;          /* initially 0 *
boolean flag[2];   /* initially false */

while(true){

      flag[i] = True;
      turn = j;
      while(flag[j] || turn[j] == j)
        ; // do nothing

      // critical-section

     flag[i] = false;
     // remainder section

}
```

ชื่อ..... รหัสนักศึกษา.....

Question 10: (25 points) List questions that you would like to see in this exam and give the answers to your questions. If you have more than one question, please assign the score to each of your questions. However, the total point can not exceed 25 points.
เขียนโจทย์ที่ท่านคาดหวังว่าจะเจอในข้อสอบชุดนี้แต่ไม่เจอ กรุณาเขียนโจทย์และตอบคำถามของท่านเองด้วย หากท่านมีโจทย์มากกว่าหนึ่งข้อ กรุณากำหนดคะแนนให้โจทย์แต่ละข้อด้วย (แต่ต้องรวมกันแล้วไม่เกิน 25 คะแนน)