ชื่อ..... รหัสนักศึกษา.....

PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Midterm Examination: 1st Semester | Academic Year: 2013

Date: 28 July 2013 | Time: 13:30-15:30

Subject: 242-304 Computer Operating Systems | Room: A200, R201

ทุจริตในการสอบ โทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

Instruction:

- Please write your name and student id on all pages. There are 12 pages (including this page).
- This examination has 6 questions (250 points). Please answer all questions in the space provided. Your answer can be in Thai. There is one extra credit question on the last page which is an optional question.
- If you have any question on the examination, please write down your assumption and continue your work according to your assumption, we will discuss the question in class.
- No book, No note, No calculator, No cell-phone

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 12 หน้า (รวมปก)
- ข้อสอบมี 6 ข้อ (250 คะแนน) กรุณาตอบทุกข้อ ในพื้นที่ที่กำหนดไว้ คุณสามารถตอบเป็นภาษาไทยได้ ข้อสอบพิเศษในหน้าสุดท้ายนั้นคุณจะตอบหรือไม่ก็ได้
- หากคุณมีปัญหาเกี่ยวกับข้อสอบ กรุณาเขียนสมมติฐานของท่านลงในข้อสอบ และทำข้อสอบตามสมมติฐานของท่านต่อไป เราจะถกเรื่องคำถามของท่านในชั้นเรียน
- ไม่อนุญาต หนังสือ เอกสาร เครื่องคิดเลข โทรศัพท์เคลื่อนที่

อาจารย์จะสามารถตรวจได้เฉพาะคำตอบที่อาจารย์อ่านออกเท่านั้น หากอาจารย์อ่านคำตอบคุณไม่ออก หรือ อ่านชื่อและรหัสนักศึกษาของคุณไม่ออก คุณจะไม่ได้คะแนน

ชื่อ..... รหัสนักศึกษา.....

ข้อมูลเพื่อใช้ในการตอบคำถาม

- Starvation ปัญหาการอดอยาก ซึ่งคือการที่โปรเซสหรืองานใดๆ ต้องรอเป็นระยะเวลาอันยาวนานหรือในบางกรณีเป็นการรอแบบไม่มีที่สิ้นสุด
- Preemptive เมื่อโปรเซสได้เริ่มรันแล้ว ระบบจะสามารถหยุดการทำงานของโปรเซสดังกล่าวและเรียกทรัพยากรคืนจากโปรเซสเพื่อไปแจกจ่ายให้กับโปรเซสอื่นได้ โดยในระหว่างนั้นตัวโปรเซสที่โดนพรีเอ็มก็จะไปรออยู่ใน ready queue
- Non-preemptive เมื่อโปรเซสได้เริ่มรันแล้ว โปรเซสจะได้ใช้งานทรัพยากรไปจนกระทั่งเสร็จสิ้นการทำงาน
- Waiting time เวลาที่โปรเซสต้องรออยู่ในระบบ
- Mutual Exclusion: หมายถึงในเวลาขณะใดขณะหนึ่ง หากมีโปรเซสหนึ่งกำลังรันอยู่ใน critical section โปรเซสอื่นๆ จะไม่สามารถรันใน critical section ได้
- Progress: หมายถึง หากไม่มีโปรเซสใดกำลังรันอยู่ใน critical section และมีโปรเซสอื่นๆ ที่กำลังรอจะเข้าไปรันใน critical section แล้วนั้น การเลือกโปรเซสเพื่อเข้าไปรันใน critical section จะไม่สามารถเลื่อนออกไปได้
- Bounded waiting: หมายถึง มีการกำหนดจำนวนครั้งหรือจำนวนโปรเซสอื่นซึ่งเข้าไปรันใน critical section ในระหว่างที่โปรเซสหนึ่งรอขอเข้าไปรันใน critical section จนถึงเวลาที่โปรเซสนั้นได้รับอนุญาตให้รันได้
- Atomic: อะตอมมิก คือ การกระทำงานซึ่งจะต้องทำให้เสร็จสิ้น หรือไม่ทำเลย มีแค่สองกรณีเท่านั้น

ชื่อ..... รหัสนักศึกษา.....

Question 1: (25 points) [10 mins] According to the status of the process in the system, please answer the following questions. จากข้อมูลสถานะของโปรเซสในระบบ จงตอบคำถามต่อไปนี้

1.1) (15 points) How many statuses can a RUNNING process change to? What are they and how?
โปรเซสจะเปลี่ยนจากสถานะดำเนินการไปเป็นสถานะได้กี่สถานะ อะไรบ้าง อย่างไร

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

1.2) (10 points) When will the process change its status from WAITING to READY?
โปรเซสจะเปลี่ยนจากสถานะรอเป็นสถานะพร้อมก็ต่อเมื่อ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

Question 2: (20 points) [10 mins] According to the program below, what would be the answer? Explain จากโปรแกรมที่แสดงข้างล่าง ผลลัพธ์คืออะไร โปรดอธิบายคำตอบของท่าน

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

int value = 20;
void *childwork(void *param);

int main(int argc, char *argv[]) {
  pthread_t tid;
  pthread_attr_t attr;
  int pid;
  value++;
  pid = fork();
  value++;
  if (pid == 0) {
    pthread_attr_init(&attr);
    pthread_create(&tid,&attr,childwork,NULL);
    pthread_join(tid,NULL);
    printf("CHILD: value = %d\n",value);
  }
  else if(pid > 0) {
    wait(NULL);
    value++;
    printf("PARENT: value = %d\n",value);
  }
}

void *childwork(void *param) {
  value = 15;
  exit(0);
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

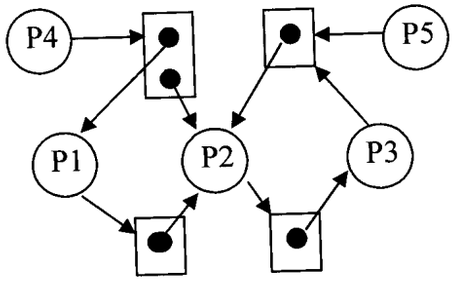
.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

Question 3: (50 points) [20 mins] Consider the following resource allocation graph, please answer whether the system is in the deadlock state or not? What processes are in the deadlock situation? What processes will you select as a victim to be restarted? Please give a reason on why you select the process.



พิจารณากราฟการแจกจ่ายทรัพยากรต่อไปนี้ จงตอบว่าระบบอยู่ในภาวะติดตายหรือไม่ โปรเซสใดที่ติดอยู่ในภาวะติดตาย คุณจะเลือกให้โปรเซสใดต้องเริ่มทำงานใหม่ จงให้เหตุผลว่าเพราะอะไรคุณจึงเลือกโปรเซสดังกล่าว

ชื่อ..... รหัสนักศึกษา.....

Question 4: (30 points) [20 mins] Please consider the following statement and select TRUE or FALSE, if the answer is FALSE, you must give your explanation
โปรดพิจารณาข้อความต่อไปนี้และเลือกว่าข้อความต่อไปนี้ ถูก หรือ ผิด หากตอบผิดกรุณาอธิบายด้วย

4.1 TRUE FALSE Multithread program can be executed on the machine with multi-core architecture only. Single-core machine cannot execute a multithread program.
โปรแกรมมัลติเธรดสามารถประมวลผลบนเครื่องที่ใช้สถาปัตยกรรมแบบมัลติคอร์เท่านั้น เครื่องแบบคอร์เดียวไม่สามารถประมวลผลโปรแกรมมัลติเธรดได้

Explanation
คำอธิบาย .....
.....
.....
.....
.....

4.2 TRUE FALSE Shortest job first scheduling algorithm could not result in starvation.
ขั้นตอนวิธีจัดกำหนดการแบบงานสั้นก่อนไม่มีปัญหาความอดอยาก

Explanation
คำอธิบาย .....
.....
.....
.....
.....

4.3 TRUE FALSE Deadlock prevention technique is not practical because it will require the system to check for a possible deadlock situation on every request.
เทคนิคการป้องกันภาวะติดตายไม่เหมาะสมเนื่องจากเทคนิคดังกล่าวทำให้ระบบต้องตรวจสอบสถานะติดตายของระบบทุกครั้งที่มีการร้องขอทรัพยากร

Explanation
คำอธิบาย .....
.....
.....
.....
.....

4.4 TRUE FALSE If the acquire() and release() semaphore operations are not executed atomically, then mutual exclusion may be violated.
หากตัวดำเนินการเซมมะฟอร์อควายและรีลีสไม่ประมวลผลแบบอะตอมมิวแล้วมูซัลล์เอกซ์ครูชันอาจถูกละเมิดได้

Explanation
คำอธิบาย .....
.....
.....
.....
.....

4.5 TRUE FALSE Remote Procedure Call (RPC) enables device controllers to inform the CPU that it has finished its operation.
อาร์พีซี ทำให้ตัวควบคุมอุปกรณ์บอกกับหน่วยประมวลผลว่าตนเองนั้นทำงานเสร็จแล้ว

Explanation
คำอธิบาย .....
.....
.....
.....
.....

4.6 TRUE FALSE The command to read a system clock must be a privilege command.
คำสั่งในการอ่านค่าเวลาของระบบต้องเป็นคำสั่งแบบพริฟวะลิจด์

Explanation
คำอธิบาย .....
.....
.....
.....
.....

ชื่อ..... รหัสนักศึกษา.....

Question 5: (65 points) [30 mins] Consider the following set of processes, with the length of the CPU burst given in milliseconds. The processes are assumed to have arrived in the order P1, P2, P3, P4, and P5. Answer the following questions

Process	Burst Time	Arrival Time
P1	7	0
P2	1	4
P3	2	4
P4	1	5
P5	5	5

จากข้อมูลของโปรเซสทั้งห้าตัวที่กำหนดให้ข้างต้น สมมุติให้โปรเซสทั้งหมดเข้ามาในระบบตามลำดับ P1, P2, P3, P4, และ P5 จงตอบคำถามต่อไปนี้

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 (both preemptive and non-preemptive), and Round-Robin (time quantum = 3). Use FCFS to break tie.
จงวาด Gantt charts เพื่อแสดงการทำงานของโปรเซสแต่ละตัว เมื่อเราใช้ขั้นตอนวิธีมาก่อนบริการก่อน งานสั้นก่อน (ทั้งแบบพรีเอมพ์ทีฟและนอนพรีเอมพ์ทีฟ) และราวน์ดโรบิน (ไทม์ควานทัม = 3) หากมีตัวเลือกสองตัวให้เลือกทำงานโปรเซสที่เข้ามาก่อนเสมอ

First-Come-First-Served (มาก่อนบริการก่อน)

Preemptive Shortest-Job-First (พรีเมพ์ทีฟงานสั้นก่อน)

ชื่อ..... รหัสนักศึกษา.....

Non-preemptive Shortest-Job-First (นอนพรีเมพ์ทีฟงานสั้นก่อน)

Round-Robin with time quantum = 3 (ราวน์ดโรบินโดยใช้ไทม์ควานทัม = 3)

b) (25 points) What is the waiting time of each process for each of these scheduling algorithms?
จงกรอกข้อมูลค่าเวลารอของโปรเซสแต่ละตัว เมื่อเราใช้ขั้นตอนวิธีการจัดกำหนดการแต่ละวิธี

Algorithm	Waiting time (milliseconds)					Average waiting time (milliseconds)
	P1	P2	P3	P4	P5	
First-come-First-Served						
Preemptive Shortest-Job-First						
Non-preemptive Shortest-Job-First						
Round-Robin						

ชื่อ..... รหัสนักศึกษา.....

Question 6: (60 points) [30 mins] Consider the solution to the producer-consumer problem below, you have to answer with an adequate explanation whether the solution can solve the problem or not? พิจารณาวิธีการแก้ปัญหาผู้ผลิตและผู้บริโภคต่อไปนี้ คุณต้องตอบพร้อมคำอธิบายที่เพียงพอว่าวิธีการแก้ปัญหานี้สามารถแก้ปัญหาได้หรือไม่

```
int N = 0;
semaphore S = 1;
semaphore Y = 0;
```

```
Producer {
    while(true){
        S.acquire();
        // Add item to buffer
        N++;
        if(N == 1)
            Y.release();
        S.release();
    }
}
```

```
Consumer{
    Y.acquire();
    while(true){
        S.acquire();
        // Take item
        N--;
        S.release();
        if(N==0)
            Y.acquire();
    }
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....