ชื่อ..... รหัสนักศึกษา.....

PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Midterm Examination: 1st Semester | Academic Year: 2012

Date: 30 July 2012 | Tme: 9:00-12:00

Subject: 241-304 Computer Operating Systems | Room: หัวหุ่นยนต์, A400, R201

Instruction:

- Closed books, Closed notes, No calculator, No computer or laptop
- Please write your name and student id on all pages. There are 13 pages.
- This examination has 8 questions. Please answer all questions. Your answer can be in Thai.
- The total score is 300.
- There is one optional question as extra credits.
- Definitions of some terms are given on the last page

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. If you have any question on any of the exam question, please write down your assumption and continue your work. No need to ask any question during the exam. We will discuss any concern in class later.

คำสั่ง

- ห้ามนำหนังสือ หรือ เอกสารใดๆ หรือเครื่องคิดเลข หรือเครื่องคอมพิวเตอร์ เข้าห้องสอบ
- กรุณาเขียนชื่อและรหัสนักศึกษาบนข้อสอบทุกหน้า ข้อสอบมีทั้งหมด 13 หน้า
- ข้อสอบมี 8 ข้อ กรุณาตอบทุกข้อ คุณสามารถตอบเป็นภาษาไทยได้
- คะแนนเต็ม 300
- มีข้อสอบพิเศษหนึ่งข้อซึ่งคุณสามารถเลือกทำหรือไม่ทำก็ได้
- คำนิยามส่วนหนึ่งได้ให้ไว้ในข้อสอบหน้าสุดท้าย

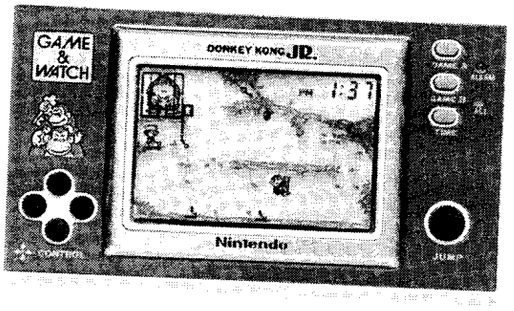
อาจารย์จะสามารถตรวจได้เฉพาะคำตอบที่อาจารย์อ่านออกเท่านั้น หากอาจารย์อ่านคำตอบคุณไม่ออก หรือ อ่านชื่อและรหัสนักศึกษาของคุณไม่ออก คุณจะไม่ได้คะแนน

หากคุณมีคำถามหรือข้อสงสัยเกี่ยวกับข้อสอบข้อใดๆ ก็ตาม ให้เขียนสมมติฐานของคุณลงในข้อสอบข้อนั้นๆ แล้วเขียนคำตอบ โดยไม่จำเป็นต้องถามคำถามระหว่างการสอบ หากมีประเด็นอาจารย์จะนำไปถกกันในชั้นเรียน

ทุจริตในการสอบ โทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

ชื่อ..... รหัสนักศึกษา.....

Question 1: (30 points) [15 minutes] Give two reasons with explanations why a simple device such as a handheld electronic game (similar to the one I have shown you in class) probably contains an OS



จงบอกเหตุผล 2 ประการพร้อมคำอธิบายว่า เพราะอะไรระบบที่ไม่มีความซับซ้อนมากมาย เช่น เครื่องเล่นเกมอันเล็กๆ (คล้ายกับอันที่อาจารย์ให้คุณดูในห้องเรียน) ดังแสดงในรูป จึงมีระบบปฏิบัติการ

ชื่อ..... รหัสนักศึกษา.....

Question 2: (60 points) [40 minutes] Answer the following questions. จงตอบคำถามต่อไปนี้

2.1 When the process is created in the system and in what state it will be in?
โพรเซสถูกสร้างขึ้นมาในระบบเมื่อไหร่และโพรเซสจะเริ่มต้นที่สถานะใด?

.....

.....

.....

.....

.....

2.2 How many states can a process change from the running state and what sort of events can cause these transitions?
โพรเซสสามารถเปลี่ยนจากสถานะทำงาน (running) ไปเป็นสถานะใดได้กี่สถานะและเหตุการณ์อะไรที่ทำให้เกิดการเปลี่ยนแปลงดังกล่าว

.....

.....

.....

.....

.....

2.3 Why most thread libraries provide no-busy waiting semaphores? What problem can it cause?
เพราะเหตุใดไลบรารีเธรดส่วนใหญ่จะบริการเซมมะฟอร์แบบ no-busy waiting และการกระทำดังกล่าวจะสร้างปัญหาอะไร

.....

.....

.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

2.4 Is it possible to have a deadlock involving only one single process? Explain your answer
เป็นไปได้หรือไม่ที่จะมีสภาวะติดตาย (deadlock) โดยมีเพียงโพรเซสเดียว อธิบาย

.....

.....

.....

.....

.....

2.5 How does an application ask the OS to do something?
โปรแกรมคอมพิวเตอร์ทั่วไปร้องขอบริการจากระบบปฏิบัติการอย่างไร

.....

.....

.....

.....

.....

2.6 What is the relationship between processes and threads?
จงอธิบายความสัมพันธ์ระหว่างโพรเซสกับเธรด

.....

.....

.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

Question 3: (30 points) [15 minutes] Explain how a multithread program can be executed on a single-core processor system?

จงอธิบายว่าโปรแกรมแบบหลายเธรดสามารถประมวลผลบนระบบที่มีหน่วยประมวลผลเดียวได้อย่างไร

ชื่อ..... รหัสนักศึกษา.....

Question 4: (65 points) [40 minutes]

Consider the following snapshot of a system. พิจารณาภาพของระบบต่อไปนี้

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 2	1 5 1 0
P1	1 0 1 0	1 7 5 0	
P2	1 3 3 4	2 3 3 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 2 5 6	

Answer the following questions using the banker's algorithm. แล้วตอบคำถามต่อไปนี้โดยใช้อัลกอริทึม banker

a. (20 points) What is the context of the matrix Need? ข้อมูลในเมทริก Need คืออะไร

	A	B	C	D
P0				
P1				
P2				
P3				
P4				

b. (15 points) Is the system in a safe state? Explain ระบบในขณะนี้ safe หรือไม่ อธิบาย

.....

.....

.....

.....

.....

.....

.....

c. (15 points) According to the system show in the Table, if a request from process P4 arrives for (0,2,1,0) can the request be granted immediately? Explain

อ้างอิงถึงระบบที่แสดงในตารางจากโจทย์ หากโพรเซส P4 ร้องขอทรัพยากร ดังต่อไปนี้ (0,2,1,0) การร้องขออันนี้สมควรจะได้รับการอนุญาตทันทีหรือไม่ อธิบาย

.....

.....

.....

.....

.....

.....

.....

.....

d. (15 points) According to the starting system show in the Table, if a request from process P3 arrives for (0,0,0,2) can the request be granted immediately? Explain

อ้างอิงถึงระบบที่แสดงในตารางจากโจทย์ตั้งต้น หากโพรเซส P3 ร้องขอทรัพยากร ดังต่อไปนี้ (0,0,0,2) การร้องขออันนี้สมควรจะได้รับการอนุญาตทันทีหรือไม่ อธิบาย

.....

.....

.....

.....

.....

.....

.....

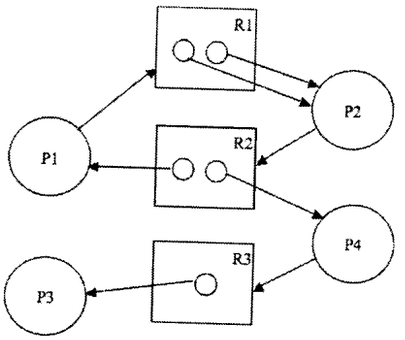
.....

.....

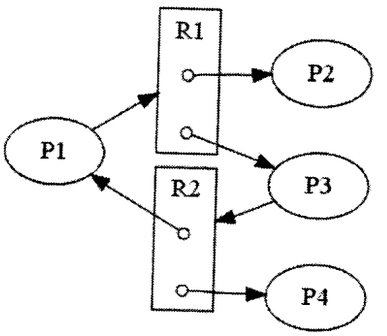
.....

ชื่อ..... รหัสนักศึกษา.....

Question 5: (25 points) [15 minutes]



(a)



(b)

From the two resource allocation graph (on your left), explain whether each situation is in a deadlock or not. If it is not in the deadlock, give the sequence of processes to be executed in order to finish all processes in the system. Assuming there is no more processes in the system. If it is in the deadlock situation, select one victim process to be terminated and give the reason why you pick that process.

จากภาพแสดงกราฟการแจกจ่ายทรัพยากรทั้งสอง อธิบายว่าในแต่ละสถานการณ์อยู่ในสภาวะติดตายหรือไม่ หากไม่อยู่ในสภาวะติดตายให้บอกลำดับการทำงานของโพรเซสเพื่อให้ทุกโพรเซสสามารถทำงานได้สำเร็จ (สมมุติว่าไม่มีโพรเซสใหม่ในระบบ) หากอยู่ในสภาวะติดตายให้เลือกโพรเซสหนึ่งตัวในระบบเป็นเหยื่อในการโดนหยุดการทำงานและให้เหตุผลว่าทำไมจึงเลือกโพรเซสดังกล่าว

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

Question 6: (40 points) [25 minutes] What is the result of the following program? Is the result of the two lines the same? justify your answer briefly

ผลลัพธ์ของโปรแกรมข้างล่างคืออะไร ผลลัพธ์ของทั้งสองบรรทัดเหมือนกันหรือไม่ อธิบายคำตอบของคุณพอสังเขป?

```
#include <pthread.h>
#include <stdio.h>
int value = 30;
void *runner(void *param);
int main(int argc, char *argv[]) {
 pthread_t tid;
 pthread_attr_t attr;
 int pid;
 pthread_attr_init(&attr);
 pthread_create(&tid,&attr,runner,NULL);
 pthread_join(tid,NULL);
 pid = fork();
 if (pid == 0) {
  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) {
 if (value%2==0)  value = 15;
 else value = value + 1;
 pthread_exit(0);
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

Question 7: (30 points) [15 minutes] Professor Vector said that the shared memory IPC is the best way to use for all communications between processes. Is Professor Vector correct? Justify your answer briefly
ศาสตราจารย์เวคเตอร์กล่าวว่าการใช้การติดต่อระหว่างโพรเซสแบบใช้หน่วยความจำร่วมกัน (shared memory IPC) เป็นวิธีการติดต่อระหว่างโพรเซสที่ดีที่สุดสำหรับการติดต่อระหว่างโพรเซส คุณคิดว่าศาสตราจารย์เวคเตอร์กล่าวถูกต้องหรือไม่ อธิบายคำตอบของคุณพอสังเขป

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

Question 8: (20 points) [15 minutes] Show that, if the acquire() and release() semaphore operations are not executed atomically, then mutual exclusion may be violated.

จงแสดงให้เห็นว่าหากคำสั่ง acquire() และคำสั่ง release() ของเซมมะฟอร์ไม่ทำงานแบบอะตอมมิกแล้วอาจทำให้เกิดปัญหากับ mutual exclusion ได้

ชื่อ..... รหัสนักศึกษา.....

Optional Extra Credits (maximum 10 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 cannot exceed 10 points. I will give points to this question according to how good your question is and the correctness of your answers

เขียนโจทย์ที่ท่านคาดหวังว่าจะเจอในข้อสอบชุดนี้แต่ไม่เจอ กรุณาเขียนโจทย์และตอบคำถามของท่านเองด้วย หากท่านมีโจทย์มากกว่าหนึ่งข้อ กรุณากำหนดคะแนนให้โจทย์แต่ละข้อด้วย (แต่ต้องรวมกันแล้วไม่เกิน 10 คะแนน) อาจารย์จะให้คะแนนคำตอบข้อนี้ของคุณโดยการพิจารณาจากคุณภาพของคำถามของคุณและความถูกต้องของคำตอบที่คุณให้

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ชื่อ..... รหัสนักศึกษา.....

Banker's Algorithm

Let n = number of processes, and m = number of resources types

Data structure

Available: Vector of length m. If available $[j] = k$, there are k instances of resource type R_j available.
Max: n x m matrix If $Max\ [i,j] = k$, then process P_i may request at most k instances of resource type R_j.
Allocation: n x m matrix If $Allocation[i,j] = k$ then P_i is currently allocated k instances of R_j.
Need: n x m matrix. If $Need[i,j] = k$, then P_i may need k more instances of R_j to complete its task.

Safety algorithm

1. Let *Work* and *Finish* be vectors of length m and n, respectively.
   Initialize:
   $Work = Available$
   $Finish\ [i] = false$ for $i = 0, 1, \ldots, n-1$.
2. Find an i such that both:
   $Finish\ [i] = false$
   $Need_i \leq Work$
   If no such i exists, go to step 4.
3. $Work = Work + Allocation_i$
   $Finish[i] = true$
   go to step 2.

If $Finish\ [i] == true$ for all i, then the system is in a safe state

Definition (คำนิยาม)

- Mutual Exclusion: หมายถึงในเวลาขณะใดขณะหนึ่ง หากมีโพรเซสหนึ่งกำลังประมวลผลอยู่ในส่วนวิกฤต (critical section) โพรเซสอื่นๆ จะไม่สามารถประมวลผลในส่วนวิกฤตของตนได้
- Progress: หมายถึง หากไม่มีโพรเซสใดกำลังประมวลผลอยู่ในส่วนวิกฤต และมีโพรเซสอื่นกำลังรอจะเข้าไปประมวลผลในส่วนวิกฤตแล้วนั้น จะต้องมีโพรเซสใดโพรเซสหนึ่งได้รับเลือกเพื่อเข้าไปประมวลผลในส่วนวิกฤตได้โดยไม่มีการเลื่อนออกไป
- Bounded waiting: หมายถึง มีการจำกัดจำนวนครั้งหรือจำนวนโพรเซสอื่นซึ่งเข้าไปประมวลผลในส่วนวิกฤต ในระหว่างที่โพรเซสหนึ่งรอขอจะเข้าไปประมวลผลในส่วนวิกฤตของตนเอง จนกระทั่งถึงเวลาที่โพรเซสนั้นได้รับอนุญาตให้เข้าไปประมวลผลในส่วนวิกฤตของตนได้
- Atomic: อะตอมมิก คือ การทำงานซึ่งจะต้องทำให้เสร็จสิ้นทั้งกระบวนการ (ชุดคำสั่ง) หรือไม่ทำเลย ทั้งนี้หากกระบวนการใดมีคำสั่งมากกว่าหนึ่งคำสั่ง ชุดคำสั่งของกระบวนการดังกล่าวจำเป็นต้องมองเสมือนเป็นก้อนเดียว กล่าวคือ คำสั่งทั้งชุดจะต้องถูกประมวลผลให้เสร็จสิ้นทั้งหมดโดยไม่มีการขัดจังหวะ หากไม่สามารถทำให้เสร็จสิ้นได้ก็จำเป็นต้องรอลแบล็คเสมือนไม่ได้ดำเนินการชุดคำสั่งนั้นเลย
- Busy-waiting semaphore คือการที่โพรเซสรอให้เงื่อนไขอันหนึ่งเป็นจริงโดยการวนรอบตรวจสอบเงื่อนไขนั้น โดยไม่ปล่อยการถือครองหน่วยประมวลผลในระหว่างรอ
- โพรเซสใช้ไอพีซี (IPC: Interprocess communication) ในการติดต่อกับโพรเซสอื่นเพื่อแลกเปลี่ยนข้อมูลกัน ซึ่งไอพีซีมีด้วยกันสองลักษณะคือ เมสเซสพาส์ซิง (Message passing) และ แชร์เมโมรี (shared memory)