

PROGRAMACION CONCURRENTES - EXAMEN FINAL - 5-8-2009

En todos los casos, responda con claridad y sintéticamente. En los casos que corresponda, NO SE CONSIDERARAN RESPUESTAS SIN JUSTIFICAR. Tiempo Máximo 2 hs 15 min.

1. Sea “ocupados” una variable entera inicializada en N que representa la cantidad de slots ocupados de un buffer, y sean P1 y P2 dos programas que se ejecutan de manera concurrente, donde cada una de las instrucciones que los componen son atómicas.

P1:: if (ocupados < N) then begin buffer := elemento_a_agregar; ocupados := ocupados + 1; end;	P2:: if (ocupados > 0) then begin ocupados := ocupados - 1; elemento_a_sacar:= buffer; end;
--	---

El programa funciona correctamente para asegurar el manejo del buffer? Si su respuesta es afirmativa justifique. Sino, encuentre una secuencia de ejecución que lo verifique y escríbala, y además modifique la solución para que funcione correctamente (Suponga buffer, elemento_a_agregar y elemento_a_sacar variables declaradas).

2. Defina el concepto de granularidad. Qué relación existe entre la granularidad de programas y de procesadores?

3. Sea la siguiente solución propuesta al problema de asignación LJM (Longest Job Next):

```
monitor LJM {  
  bool libre = true;  
  cond turno;  
  procedure request(int tiempo) {  
    if (not libre) wait(turno, (MAXVALOR - tiempo));  
    libre = false;  }  
  procedure release() {  
    libre = true;  
    signal(turno);  }  
}
```

- a) Funciona correctamente con disciplina de señalización Signal and Continue? Justifique.
b) Funciona correctamente con disciplina de señalización Signal and Wait? Justifique

Nota: suponga que MAXVALOR es el máximo valor de tiempo por el cual un proceso puede solicitar el recurso.

4. a) Qué significa que un problema sea de “exclusión mutua selectiva”?
b) El problema de los lectores–escritores es de exclusión mutua selectiva? Por qué?
c) De los problemas de los baños planteados en teoría, cuál podría ser de exclusión mutua selectiva? Por qué?
d) Por qué el problema de los filósofos es de exclusión mutua selectiva? Si en lugar de 5 filósofos fueran 3, el problema seguiría siendo de exclusión mutua selectiva? Por qué?
e) El problema de los filósofos resuelto en forma centralizada y sin posiciones fijas es de exclusión mutua selectiva? Por qué?
f) Si en el problema de los lectores–escritores se acepta sólo 1 escritor o 1 lector en la BD, tenemos un problema de exclusión mutua selectiva? Por qué?

5. Resuelva con monitores el siguiente problema:

Tres clases de procesos comparten el acceso a una lista enlazada: *searchers*, *inserters* y *deleters*. Los *searchers* sólo examinan la lista, y por lo tanto pueden ejecutar concurrentemente unos con otros. Los *inserters* agregan nuevos items al final de la lista; las inserciones deben ser mutuamente exclusivas para evitar insertar dos items casi al mismo tiempo. Sin embargo un insert puede hacerse en paralelo con uno o más *searchers*. Por último, los *deleters* remueven items de cualquier lugar de la lista. A lo sumo un deleter puede acceder la lista a la vez, y el borrado también debe ser mutuamente exclusivo con *searchers* e inserciones.

6. Explique sintéticamente los 7 paradigmas de interacción entre procesos en programación distribuida. Ejemplifique en cada caso.

7. a) Cuál es el objetivo de la programación paralela?

b) Defina las métricas de speedup y eficiencia. Cuál es el significado de cada una de ellas (que miden)? Ejemplifique.

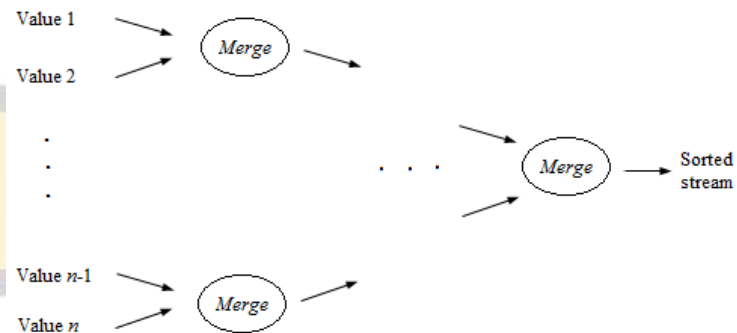
c) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 1000 unidades de tiempo, de las cuales el 80% corresponden a código paralelizable. Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo?

8. Suponga los siguientes métodos de ordenación de menor a mayor para n valores (n par y potencia de 2), utilizando pasaje de mensajes:

i- Un pipeline de filtros. El primero hace input de los valores de a uno por vez, mantiene el mínimo y le pasa los otros al siguiente. Cada filtro hace lo mismo: recibe un stream de valores desde el predecesor, mantiene el más chico y pasa los otros al sucesor.

ii- Una red de procesos filtro (como la de la figura).

iii- Odd/even exchange sort. Hay n procesos $P[1:n]$. Cada uno ejecuta una serie de rondas. En las rondas "impares", los procesos con número impar $P[\text{impar}]$ intercambian valores con $P[\text{impar}+1]$. En las rondas "pares", los procesos con número par $P[\text{par}]$ intercambian valores con $P[\text{par}+1]$ ($P[1]$ y $P[n]$ no hacen nada en las rondas "pares"). En cada caso, si los números están desordenados actualizan su valor con el recibido.



Asuma que cada proceso tiene almacenamiento local sólo para dos valores (el próximo y el mantenido hasta ese momento).

a) Cuántos procesos son necesarios en i e ii? Justifique.

b) Cuántos mensajes envía cada algoritmo para ordenar los valores? Justifique.

c) En cada caso, cuáles mensajes pueden ser enviados en paralelo (asumiendo que existe el hardware apropiado) y cuáles son enviados secuencialmente? Justifique.

d) Cuál es el tiempo total de ejecución de cada algoritmo? Asuma que cada operación de comparación o de envío de mensaje toma una unidad de tiempo. Justifique.