

PROGRAMACION CONCURRENTE - EXAMEN FINAL 13-10-2010

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. Cómo pueden clasificarse las arquitecturas multiprocesador.
 - a) según el mecanismo de control? Describa.
 - b) según la organización del espacio de direcciones? Describa
 - c) según la red de interconexión? Describa.
2. a) Qué significa el problema de “interferencia” en programación concurrente? Cómo puede evitarse?
b) En qué consiste la propiedad de “A lo sumo una vez” y qué efecto tiene sobre las sentencias de un programa concurrente? De ejemplos de sentencias que cumplan y de sentencias que no cumplan con ASV.
3. En los protocolos de acceso a sección crítica vistos en clase, cada proceso ejecuta el mismo algoritmo. Una manera alternativa de resolver el problema es usando un proceso coordinador. En este caso, cuando cada proceso SC[i] quiere entrar a su sección crítica le avisa al coordinador, y espera a que éste le de permiso. Al terminar de ejecutar su sección crítica, el proceso SC[i] le avisa al coordinador.
Desarrolle protocolos para los procesos SC[i] y el coordinador usando sólo variables compartidas (no tenga en cuenta la propiedad de eventual entrada).
4. a) Defina el concepto de “sincronización barrier”.Cuál es su utilidad?
b) Qué es una barrera simétrica?
c) Describa “combining tree barrier” y “butterfly barrier”. Marque ventajas y desventajas en cada caso.
5. Dados los siguientes dos segmentos de código, indicar para cada uno de los ítems si son equivalentes o no. Justificar cada caso (de ser necesario dar ejemplos).

Segmento 1	Segmento 2
<pre>... int cant=1000; DO (cant < -10); datos?(cant) → Sentencias1 □ (cant > 10); datos?(cant) → Sentencias2 □ (INCOGNITA); datos?(cant) → Sentencias3 END DO ...</pre>	<pre>... int cant=1000; While (true) { IF (cant < -10); datos?(cant) → Sentencias1 □ (cant > 10); datos?(cant) → Sentencias2 □ (INCOGNITA); datos?(cant) → Sentencias3 END IF } ...</pre>

- a) *INCOGNITA* equivale a: *(cant = 0)*,
 - b) *INCOGNITA* equivale a: *(cant > -100)*
 - c) *INCOGNITA* equivale a: *((cant > 0) or (cant < 0))*
 - d) *INCOGNITA* equivale a: *((cant > -10) or (cant < 10))*
 - e) *INCOGNITA* equivale a: *((cant >= -10) or (cant <= 10))*
6. Explique sintéticamente los 7 paradigmas de interacción entre procesos en programación distribuida. En cada caso ejemplifique, indique qué tipo de comunicación por mensajes es más conveniente y qué arquitectura de hard que se ajusta mejor? Justifique sus respuestas.
 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 (qué miden) y su rango de valores? Ejemplifique.
- c) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup (S) está regido por la función $S=p-1$ y en el otro por la función $S=p/2$.Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores? Justifique claramente.
- d) Suponga que el tiempo de ejecución de un algoritmo secuencial es de 10000 unidades de tiempo, de las cuales sólo el 95% corresponde a código paralelizable.Cuál es el límite en mejora que puede obtenerse paralelizando el algoritmo? Justifique
8. Suponga n^2 procesos organizados en forma de grilla cuadrada. Cada proceso puede comunicarse solo con los vecinos izquierdo, derecho, de arriba y de abajo (los procesos de las esquinas tienen solo 2 vecinos, y los otros en los bordes de la grilla tienen 3 vecinos). Cada proceso tiene inicialmente un valor local v .
- a) Escriba un algoritmo heartbeat que calcule el máximo y el mínimo de los n^2 valores. Al terminar el programa, cada proceso debe conocer ambos valores. (Nota: no es necesario que el algoritmo esté optimizado).
- b) Analice la solución desde el punto de vista del número de mensajes.
- c) Puede realizar alguna mejora para reducir el número de mensajes?

La Fuente