

- 1) Defina el problema de sección crítica. Compare los algoritmos para resolver este problema (spin locks, Tie Breaker, Ticket, Bakery). Marque ventajas y desventajas de cada uno.
- 2) Resuelva el problema de acceso a Sección Crítica usando un proceso coordinador. En este caso cuando proceso SC[I] quiere entrar en su sección crítica le avisa al coordinador, y espera que este le de permiso. Al Terminar de ejecutar su sección crítica el proceso SC[I] le avisa al coordinador. Desarrolle una solución de grano fino usando solo variables compartidas (no semáforos, ni monitores).
- 3) Suponga que una imagen se encuentra representada por una matriz a ($n \times n$), y que el valor de cada pixel es un número entero que es mantenido por un proceso distinto (es decir, el valor del pixel I, J , está en el proceso $P(I, J)$). Cada proceso puede comunicarse solo con sus vecinos izquierdo, derecho, arriba y abajo. (los procesos de las esquinas tienen solo 2 vecinos, y los otros bordes de la grilla tienen 3 vecinos).
- a) Escriba un algoritmo Heatmap que calcule el máximo y el mínimo valor de los pixels de la imagen. Al terminar el programa, cada proceso debe conocer ambos valores.
 - b) Analice la solución desde el punto de vista del número de mensajes.
 - c) Puede realizar alguna mejora para reducir el número de mensajes.
- 4) ¿Cuál es el objetivo de la programación paralela?
 - a) Defina las métricas de speedup y eficiencia. ¿Cuál es el significado de cada una de ellas (qué miden)? ¿Cuál es el rango de valores posibles de cada una? Ejemplifique.
 - b) Defina escalabilidad de un sistema paralelo.
 - c) Suponga que la solución a un problema es paralelizada sobre p procesadores de dos maneras diferentes. En un caso, el speedup(5) está regido por la función $S = p - 4$ y el otro por la función $S = p/3$ para $p > 4$. ¿Cuál de las dos soluciones se comportará más eficientemente al crecer la cantidad de procesadores.
 - d) describa conceptualmente qué dice la ley de Amdahl (no es necesaria la fórmula).
 - e) suponga que el tiempo de ejecución de un algoritmo secuencial es de 10000 unidades de tiempo, de las cuales 95% corresponden a código paralelizable. ¿Cuál es el límite en la mejora que puede obtenerse paralelizando el algoritmo.
- 5) Sea la siguiente solución al problema del producto de matrices de $n \times n$ con p procesos en paralelo con variables compartidas:

```
Process worker[w=1 to p] {#strips en paralelo (p strips de n/p filas)}
Int first= (w-1) * n/p + 1
Int last= first + n/p - 1
For [i=1 to last]
{
  For [j= 1 to n]
  {
    C[i, j] = 0.0;
    For [k= 1 to N] {
      C[I,J] = C[I,J] + a [I,K] * b[K,j];
    }
  }
}
```

- a) Suponga $n = 128$ y cada procesador capaz de ejecutar un proceso. ¿Cuántas asignaciones, sumas y productos se hacen secuencialmente (caso en que $p=1$)? ¿Cuántos se realizan en cada procesador en la solución paralela con $p=4$?
- b) si $p_1=p_2=\dots=p_4$ y los tiempos de asignación son 1, de suma 2 y de producto 3, y si p_4 es 4 veces más lento, ¿cuánto tarda el proceso total concurrente? ¿Cuál es el valor del speedup? Modifique el código para lograr un mayor speedup.