



Preguntas tipo (Repaso de conceptos sobre OpenMP)

Estas preguntas están diseñadas para ayudarte a practicar los conceptos clave de OpenMP antes del examen. No es necesario escribir código, pero es importante comprender la teoría para interpretar correctamente las situaciones planteadas.

Conceptos Básicos de OpenMP

1. ¿Qué es OpenMP y para qué se utiliza?
2. ¿Qué tipo de paralelismo implementa OpenMP, a nivel de hilos o procesos?
3. ¿Qué directiva de OpenMP se utiliza para paralelizar un bucle `for`?
4. ¿Qué ocurre si no especificamos ninguna cláusula `private` o `reduction` en una variable compartida en un bucle paralelo?
5. ¿Qué significa que una variable sea `shared` en OpenMP?
6. ¿Qué función de OpenMP devuelve el número total de hilos en una región paralela?
7. ¿Qué significa el concepto de **condición de carrera**?

Sincronización

8. ¿Qué directiva de OpenMP garantiza que una sección de código sea ejecutada por un único hilo a la vez?
9. ¿Qué diferencia hay entre `critical` y `atomic`?
10. ¿Qué directiva de OpenMP impide que un hilo continúe hasta que todos los demás han llegado a un punto?
11. ¿Por qué es más eficiente `reduction` que `critical` para operaciones de suma?
12. ¿Qué ocurre si se usa una directiva `barrier` dentro de un bucle `for` paralelo?

Variables y Ámbito de Datos

13. ¿Qué diferencia hay entre `private` y `firstprivate`?
14. ¿Qué finalidad tiene `lastprivate`?
15. ¿Qué función tiene la cláusula `copyin`?
16. ¿Qué pasa si una variable no se declara `private` ni `shared`?

Balanceo de Carga y Schedule

17. ¿Qué función tiene la cláusula `schedule` en un bucle paralelo?



18. ¿Qué tipo de `schedule` conviene utilizar si las iteraciones tienen distinta carga de trabajo?
19. ¿Qué `schedule` ofrece bloques de trabajo decrecientes para mejorar el balanceo de carga?
20. ¿Qué impacto puede tener elegir un `chunk size` demasiado pequeño en un `schedule(dynamic)`?
21. ¿Qué se entiende por **desbalanceo de carga** en el contexto de OpenMP?

Análisis de Rendimiento

22. ¿Qué es el **Speed-up** y cómo se calcula?
23. ¿Qué es la **Eficiencia** y qué valores indican un buen rendimiento?
24. ¿Qué es la **Escalabilidad Fuerte**?
25. ¿Qué es la **Escalabilidad Débil**?
26. ¿Qué representa la Ley de Amdahl en la programación paralela?
27. ¿Qué implica que un código tenga una fracción no paralelizable del 20%?
28. ¿Qué se espera que ocurra con la eficiencia al aumentar el número de hilos en un sistema real?
29. ¿Qué factores pueden hacer que el Speed-up real sea inferior al teórico?
30. ¿Qué significa que un programa tenga un Speed-up de 1 en OpenMP?