

BP3Solucion.pdf



FaReLiLoCa



Arquitectura de Computadores



2º Grado en Ingeniería Informática

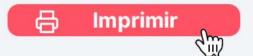


Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación Universidad de Granada





Lo que faltaba en Wuolah





Preguntas: 10

Respuestas válidas:

D

Puntuación: 1

> Nota: D

¿Con cuántas hebras se ejecuta este código si previamente se ha fijado la variable de entorno OMP_NUM_THREADS=8?

Elección única

```
omp_set_num_threads(4);
#pragma omp parallel num_threads(2)
printf("hello\n");
```

Usuario Profesores

D a) 2

D b) 1

D c) 4

1 d) 8

¿Cuál de las siguientes formas es la correcta para fijar a 4 el número de hebras para un programa OpenMP ?

Elección única

Usuario Profesores

a) En un progarma OpenMP, usando la función 1 omp_max_threads(4) al principio de la función main.

10 b) En un programa OpenMP, usando la función omp_set_num_threads(4) al principio de la función main.

1 c) En un programa OpenMP, usando la función omp_num_threads(4) al principio de la función main.

D d) En la consola del sistema, usando la variable de entorno export OMP_THREAD_LIMIT=4

El parámetro chunk en el siguiente código determina: #pragma omp parallel for schedule(guided,chunk) Elección única

Usuario Profesores

a) El tamaño del bloque iteraciones que OpenMP asignará 1



siempre a cada hebra

- b) El tamaño del bloque iteraciones óptimo que OpenMP debe usar para minimizar el tiempo de ejecución
- c) El tamaño mínimo del bloque iteraciones que OpenMP asignará a una hebra
 - d) El tamaño máximo del bloque iteraciones que OpenMP asignará a una hebra

Indica qué reparto de iteraciones a hebras es correcto suponiendo 3 hebras y la cláusula schedule(dynamic,2).

Elección única

Usuario Profesores

(D)	a)	iteración	0	1	2	3	4	5	6	7	8	9
		hebra	0	0	1	1	2	2	0	0	0	2

5 Elección única

¿Qué muestra la ejecución del siguiente programa por pantalla suponiendo que se ejecuta en un nodo de ATCgrid?

int n = (int)(omp_get_max_threads() / 4);

#pragma omp parallel for num_threads(6) if(n > 6)

for (int i=0 ; i<n ; i++)

printf("h: \%d ", omp_get_thread_num());

Usuario Profesores

- a) h:0 h:0 h:0 h:0 h:0 h:0
- b) Las otras respuestas no son correctas.
- © h:0
- d) h:0 h:1 h:2 h:3 h:4 h:5

Indica qué reparto de iteraciones a hebras es correcto suponiendo dos hebras y la cláusula schedule (guided, 3)

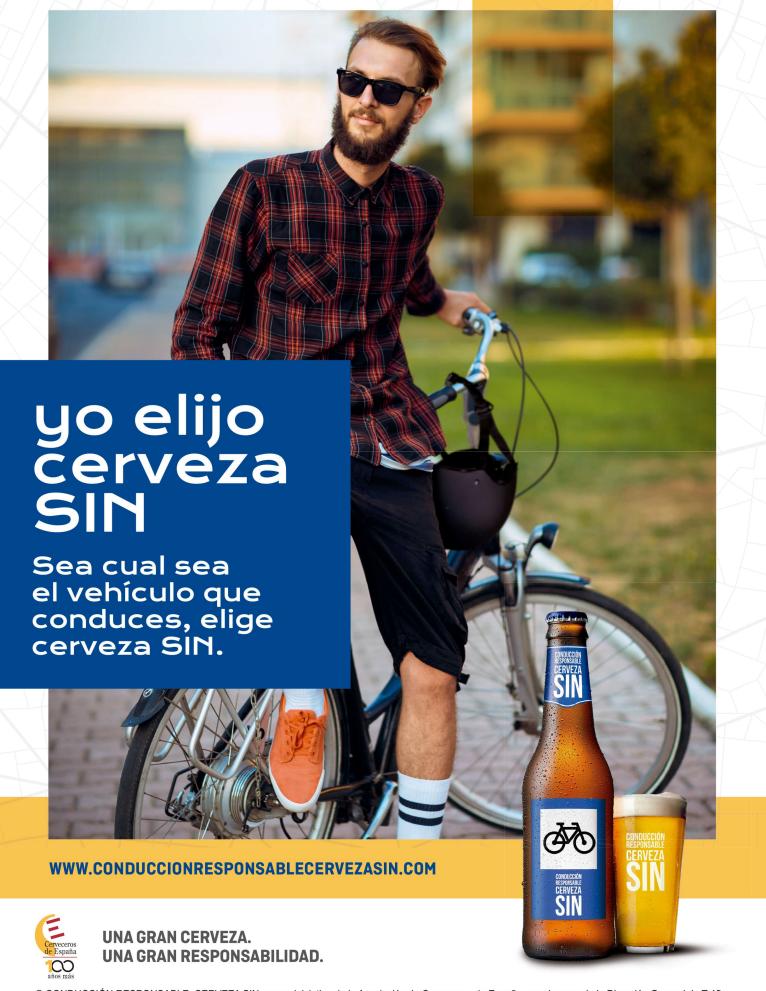
Elección única

Usuario Profesores

•	(D)	a)													
		,	iteración	0	1	2	3	4	5	6	7	8	9	10	11
			hebra	0	0	0	0	0	0	1	1	1	0	0	0

(ID)	b)												
•	íteración	0	1	2	3	4	5	6	7	8	9	10	11
	hebra	0	0	0	0	0	0	0	1	1	0	0	0

(D)	c)												
`	í iteración	0	1	2	3	4	5	6	7	8	9	10	11
	hahra	U	Λ	Λ	1	1	1	U	Λ	Λ	1	1_	1



© CONDUCCIÓN RESPONSABLE, CERVEZA SIN es una iniciativa de la Asociación de Cerveceros de España con el apoyo de la Dirección General de Tráfico.



























(D)	d)												
•	iteración	0	1	2	3	4	5	6	7	8	9	10	11
	$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	0	0	0	0	0	0	1	1	1	0	0	1

Cuál de las siguientes opciones permitiría comprobar qué tipo de planificación obtiene mejores resultados para un programa paralelo con ayuda de un *script*?

Usuario Profesores

- a) schedule(dynamic)
- b) schedule(runtime)
 - © C) schedule(static)
 - d) schedule(guided)
- Dado el código que se tiene a continuación ¿Qué tipo de reparto de iteraciones a hebras sería el más óptimo en tiempo de ejecución?

Elección única

```
#pragma omp parallel for
for (int i=0 ; i<100 ; i++)
    a[i] += b[i]*c[i];</pre>
```

Usuario Profesores

- a) static
 - b) runtime
 - c) dynamic
 - d) guided
- ¿Qué código cree mejor para conseguir multiplicar una matriz triangular inferior por un vector?

Elección única

```
int m[N][N], v[N], r[N] = {0};
Usuario Profesores
```

- - a) for (int i=0; i<N; i++)
 for (int j=0; j<=i; j++)
 - r[i] += m[i][j] * v[j];

 - © C) for (int i=0; i<N; i++)
 for (int j=i; j<=i; j++)
 - r[i] += m[i][j] * v[j];

 d) for (int j=0; j<N; j++)
 for (int i=0; i<N; i++)
 r[i] += m[i][j] * v[j];
- Indica qué reparto de iteraciones a hebras es correcto suponiendo 4 hebras y la cláusula schedule(static,3).

Elección única

Usuario Profesores

(D)	a)											
	,	iteración	0	1	2	3	4	5	6	7	8	9
		hebra	0	0	1	1	2	2	3	3	0	0

(D)	C)											
•	,	iteración	0	1	2	3	4	5	6	7	8	9
		hebra	0	0	0	0	1	1	1	1	2	2

•	(D)	d)	iteración	0	1	2	3	4	5	6	7	8	9
			hebra	0	0	0	1	1	1	2	2	2	3

