



Preguntas: 10

Respuestas
válidas: 

Puntuación: 

Nota: 





1

Elección única

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

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

Usuario Profesores





-  a) 2
-  b) 1
-  c) 4
-  d) 8

2

Elección única

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

Usuario Profesores

-  a) En un programa OpenMP, usando la función `omp_max_threads(4)` al principio de la función main.
-  b) En un programa OpenMP, usando la función `omp_set_num_threads(4)` al principio de la función main.
-  c) En un programa OpenMP, usando la función `omp_num_threads(4)` al principio de la función main.
-  d) En la consola del sistema, usando la variable de entorno `export OMP_THREAD_LIMIT=4`

3

Elección única

El parámetro `chunk` en el siguiente código determina:

```
#pragma omp parallel for schedule(guided,chunk)
```

Usuario Profesores

-  a) El tamaño del bloque iteraciones que OpenMP asignará

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

4 Indica qué reparto de iteraciones a hebras es correcto suponiendo 3 hebras y la cláusula `schedule(dynamic,2)`.
Elección única Usuario Profesores

☐ a)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	0	1	1	2	2	0	0	0	2

☐ b)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	0	0	1	1	1	2	2	2	0

• ☐ c)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	0	1	1	2	2	1	1	0	0

☐ d)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	0	1	1	2	2	2	2	0	1

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

```
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.
- ☐ c) h:0
- ☐ d) h:0 h:1 h:2 h:3 h:4 h:5

6 Indica qué reparto de iteraciones a hebras es correcto suponiendo dos hebras y la cláusula `schedule(guided,3)`.
Elección única Usuario Profesores

• ☐ 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

☐ b)

iteración	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

☐ c)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	0	0	1	1	1	0	0	0	1	1	1

hebra | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1



d)

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	1

7

Elección única

¿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)`

8

Elección única

Dado el código que se tiene a continuación ¿Qué tipo de reparto de iteraciones a hebras sería el más óptimo en tiempo de ejecución?

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

Usuario Profesores

•



a) `static`



b) `runtime`



c) `dynamic`



d) `guided`

9

Elección única

¿Qué código cree mejor para conseguir multiplicar una matriz triangular *inferior* por un vector?

```
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];
```



```
b) for (int i=0 ; i<N ; i++)
    for (int j=0 ; j<N ; 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];
```

10

Elección única

Indica qué reparto de iteraciones a hebras es correcto suponiendo 4 hebras y la cláusula `schedule(static,3)`.

Usuario Profesores



a)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	0	1	1	2	2	3	3	0	0



b)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	1	2	3	0	1	2	3	0	1



c)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	0	0	0	1	1	1	1	2	2

•



d)

iteración	0	1	2	3	4	5	6	7	8	9
hebra	0	0	0	1	1	1	2	2	2	3