



Examen BP3 Grupo 1



Universidad de Granada - Doble Grado en Ingeniería Informática y Matemáticas
Arquitectura de Computadores



Desconocido: 7256819 Muñoz Sánchez, David



Inicio: Hoy, jueves, 08:40:07

Final: Hoy, jueves, 08:57:40

Preguntas: 10

Respuestas
válidas:

Puntuación:

Nota:

1

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	1	2	3	0	1	2	3	0	1

•



b)

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



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	1	1	2	2	3	3	0	0

2

Elección única

¿Qué tipo de reparto realiza el siguiente código?

```
#pragma omp parallel for schedule(runtime)
for (i=0 ; i<1000 ; i++)
    v1[i] = v3[i] + v2[i];
```



Usuario Profesores



a) Las otras respuestas no son correctas.



b) El que indique la variable de control interno `def-sched-var`.





-  c) El que indique la variable de control interno `run-sched-var`.
-  d) El que devuelve la función de entorno `omp_set_dynamic()`.

3El 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 siempre a cada hebra
-  b) El tamaño del bloque iteraciones óptimo que OpenMP debe usar para minimizar el tiempo de ejecución
-  c) El tamaño máximo del bloque iteraciones que OpenMP asignará a una hebra
-  d) El tamaño mínimo del bloque iteraciones que OpenMP asignará a una hebra


4


Indica cuál de las siguiente opciones obtendrá mejores prestaciones para multiplicar una matriz triangular por un vector


Elección única

Usuario Profesores

-  a)

```
#pragma omp for schedule(guided)
for (i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<=i ; j++)
        v2[i] += M[i][j] * v1[j];
}
```
-  b)

```
#pragma omp for private(j) schedule(guided)
for (i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<=i ; j++) {
        #pragma omp critical
        v2[i] += M[i][j] * v1[j];
    }
}
```
-  c)





```
#pragma omp for private(j) schedule(static)
for (i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<N ; j++)
        v2[i] += M[i][j] * v1[j];
}
```
-  d)

```
#pragma omp for private(j) schedule(guided)
for (i=0 ; i<N ; i++) {
    v2[i] = 0;
    for (j=0 ; j<=i ; j++)
        v2[i] += M[i][j] * v1[j];
}
```

5¿Cómo se podría establecer el número de hebras a ejecutar en una región paralela desde `programa`?

Elección única

Usuario Profesores

-  a) Mediante el uso de la función `omp_set_num_threads` después de que una región paralela comience
-  b) Ninguna de las respuestas es correcta
-  c) Mediante el uso de la variable de entorno `OMP_NUM_THREADS` antes de la ejecución del programa
-  d) Mediante el uso de la cláusula `num_threads` en una directiva que abra la región paralela

6 ¿Cuál es la función de la cláusula `if` en el siguiente código?

Elección única

#pragma omp parallel if(n>20)

Usuario Profesores

- ☐ a) Las otras respuestas no son correctas
- ☒ b) Evitar la sobrecarga introducida al paralelizar el código para tamaños del problema pequeños
- ☐ c) No ejecutar el código del bloque estructurado si $n \leq 20$
- ☐ d) Ejecutar las ramas `if` y `else` del bloque estructurado en paralelo

7 ¿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 j=0 ; j<N ; j++)
  for (int i=0 ; i<N ; i++)
    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=0 ; j<=i ; j++)
    r[i] += m[i][j] * v[j];
```
- ☐ d)

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

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

Elección única

Usuario Profesores

- ☐ a)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	0	0	1	1	1	2	2	2	2	2	0
- ☐ b)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	0	1	1	2	2	0	0	1	1	1	1
- ☐ c)

iteración	0	1	2	3	4	5	6	7	8	9	10	11
hebra	0	1	2	2	1	2	0	1	2	0	1	2
- ☒ d)

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

9 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	1	1	1	1	2	2	2	2



b)

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

•



c)

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



d)

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

10

Elección única

¿Cuál de los siguientes métodos para determinar el número de hebras que ejecutarán la siguiente región paralela es el más prioritario?

Usuario Profesores

•

a) La cláusula `if`b) La función `omp_set_num_threads`c) La cláusula `num_threads`d) La variable de entorno `OMP_NUM_THREADS`