

13. ¿Cómo se puede modificar el reparto de iteraciones del bucle de una directiva `#pragma omp for` entre las hebras si usamos la cláusula `schedule(runtime)`?

- a) Usando la variable de retorno `OMP_SCHEDULE` Y la función `omp_set_schedule()`
- b) Usando sólo la función `omp_set_schedule()`
- c) Usando la variable de entorno `OMP_SCHEDULE` O la función `omp_set_schedule()`
- d) Usando sólo la variable de entorno `OMP_SCHEDULE`

14. Las variables de control internas de OpenMP

- a) Sólo se pueden modificar mediante el uso de variables de entorno o en la consola del sistema
- b) Las otras respuestas son todas incorrectas
- c) Sólo se pueden modificar mediante el uso de las funciones que proporciona el API de OpenMP
- d) Pueden ser accedidas directamente por el programador

15. ¿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");
```

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

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

```
Int m[N][N], v[N], r[N] = {0};
```

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

19. Las variables de control internas de OpenMP

- a) Pueden ser accedidas directamente por el programador
- b) Las otras respuestas son todas incorrectas
- c) Sólo se pueden modificar mediante el uso de las funciones que proporciona el API de OpenMP
- d) Sólo se pueden modificar mediante el uso de variables de entorno en la consola del sistema

20. Cuando se usa una planificación `dynamic` de un bucle `for` en OpenMP, el tamaño del chunk...

- a) Es siempre constante
- b) Va decreciendo conforme se van ejecutando las iteraciones del bucle

- c) Se adapta dinámicamente en función de la velocidad de cada hebra
- d) Siempre debe ser mayor que 1

21. ¿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());
```

- a) h:0
- b) las otras respuestas no son correctas
- c) h:0 h:0 h:0 h:0 h:0 h:0
- d) h:0 h:1 h:2 h:3 h:4 h:5

22. ¿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");
```

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

23. ¿Qué código cree mejor para conseguir multiplicar una matriz triangular superior por un vector?

```
Int m[N][N], v[N], r[N] = {0};
```

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

24. Si le piden que realice un estudio de escalabilidad de un código que calcula el producto de dos matrices

- a) Representaría en una gráfica el tiempo de ejecución en función del tamaño de las matrices
- b) Representaría en una gráfica la ganancia en velocidad (o ganancia en prestaciones) en función del número de núcleos
- c) No haría nada de lo indicado en el resto de respuestas
- d) Representaría en una gráfica el tiempo de ejecución en función del número de núcleos

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

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

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

c) En la consola del sistema, usando la variable de entorno `export OMP_THREAD_LIMIT=4`

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

15. En una máquina con 8 cores y tras ejecutar `export OMP_NUM_THREADS=4`, ¿cuántas iteraciones ejecuta la hebra máster en la región paralela?

```
int N = omp_get_max_threads();
```

```
omp_set_num_threads(2);
```

```
#pragma omp parallel for num_threads(6) if(N>=4) schedule(static)
```

```
for (int i=0; i <12 ; i++)
```

```
printf(" thread: %d iteración: %d\n", omp_get_thread_num(), i);
```

a) 6

b) 4

c) 12

d) 2

16. Indica que reparto de iteraciones a hebras es correcto suponiendo dos hebras y la cláusula `schedule(guided,3)`

a)

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

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	0

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	1

d)

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