**VARIABLES DE CONTROL ->** Sus valores afectan a las operaciones de las regiones paralelas.

1. **nthreads-var ->** Almacena nº threads solicitados para futuras regiones paralelas.
2. **dyn-var ->** Controla que el ajuste dinámico del nº de threads usados para futuras regiones paralelas esté activo**.**
3. **nest-var ->** Controla si el paralelismo anidado está activo para futuras regiones paralelas.
4. **run-sched-var ->** Planificación de bucles para runtime
5. **def-sched-var ->** Planificación de bucles por defecto. Ámbito de programa.

**VARIABLES DE ENTORNO ->** Usadas para modificar variables de control internas.

1. **OMP\_SCHEDULE ->** Establece run-sched-var para el tipo de runtime y el chunk (tam. segmento).
2. **OMP\_NUM\_THREADS ->** Establece nthreads-var para el numero de threads a usar en las regiones paralelas.
3. **OMP\_DYNAMIC ->** Estable dyn-var para el ajuste dinámico de los threads a utilizar para las regiones paralelas.
4. **OMP\_NESTED ->** Establece nest-var para habilitar o deshabilitar el paralelismo anidado.

**FUNCIONES PARA MODIFICAR VARIABLES DE CONTROL:**

1. **omp\_get\_dynamic () ->** Usada en el caso de que se quiera transferir el código a otro compilador compatible con OpenMP.
2. **omp\_set\_dynamic () ->** Usada en el caso de que se quiera transferir el código a otro compilador compatible con OpenMP.
3. **omp\_get\_num\_threads () ->** Obtiene el nº de threads que se están usando en una región paralela (1 en código secuencial).
4. **omp\_set\_num\_threads () ->** Especifica el nº de threads usados para las regiones paralelas.
5. **omp\_get\_thread\_num () ->** Devuelve al thread su identificador dentro del grupo del thread.
6. **omp\_get\_max\_threads () ->** Devuelve el nº de threads usados para las regiones paralelas que no especifican una cláusula num\_threads.
7. **omp\_get\_num\_procs () ->** Devuelve el nº de procesadores disponibles para el programa en el momento de la ejecución.
8. **omp\_in\_parallel () ->** Devuelve true si se llama a la rutina dentro de una región parallel activa y false en caso contrario.
9. **omp\_get\_thread\_limit ->** Devuelve el nº máximo de threads de un programa.
10. **omp\_get\_schedule (kind, modifier) ->** Devuelve el tipo de programación que se aplica al utilizar un runtime. “kind” devuelve el tipo de programa que se utiliza. “modifier” representa el tamaño del chunk que se establece.
11. **omp\_set\_schedule (kind, modifier) ->** Establece el tipo de programa a utilizar, dado por el argumento “kind”. “modifier” indica el tamaño del chunk que se va a establecer.

**ORDEN DE PRECEDENCIA PARA FIJAR Nº DE THREADS**

* nº resultante de evaluar la cláusula if
* nº que fija la cláusula num\_threads
* nº que fija la función omp\_set\_num\_threads ()
* contenido de la var. de entorno OMP\_NUM\_THREADS
* fijado por defecto por la implementación: normalmente el nº cores de un nodo aunque puede variar dinámicamente

**CLAUSULAS**

* **if ->** Si no se cumple la condición que se le indica al if, no hay ejecución paralela SOLAMENTE EN CONSTRUCCIONES CON PARALLEL.
* **num\_threads ->** indica el número de threads a utilizar.

**CLAUSULA SCHEDULE**

* SOLO FUNCIONA EN BUCLES.
* POR DEFECTO, TIPO STATIC EN LA MAYOR PARTE DE IMPLEMENTACIONES
* MEJOR NO ASUMIR UN CHUNK POR DEFECTO.

**TIPOS:**

* + **STATIC:** Las iteraciones se dividen en unidades de chunk iteraciones. Las unidades se asignan en round-robin.

La entrada es el nº de iteraciones de la unidad de distribución.

Se asigna un único chunk a cada thread.

* + **DYNAMIC:** Distribución en tiempo de ejecución. Apropiado si se desconoce el tiempo de ejecución de las iteraciones. La unidad de distribución tiene chunk iteraciones.

Nº unidades o (n/chunk).

AÑADE SOBRECARGA ADICIONAL.

Las iteraciones se dividen en unidades de chunk iteraciones.

Las unidades se asignan en tiempo de ejecución. Los threads más rápidos ejecutan más unidades.

Si no se especifica chunk, se usan unidades de asignación.

ENTRADA: nº iteraciones y tamaño unidad de distribución.

* + **GUIDED:** Distribución en t ejecución. Apropiado si se desconoce el t de ejecución de las iteraciones. Comienza con bloque largo. El tam de bloque va menguando (nº iteraciones / nº threads > chunk).

SOBRECARGA EXTRA, MENOS QUE DYNAMIC.

* + **RUNTIME:** El tipo de distribución se fija en tiempo de ejecución. Depende de la variable de control run-sched-var. Usa Schedule (runtime) y OMP\_SCHEDULE para fijar tipo de distribución.

ENTRADA: nº iteraciones.

**FUNCIONES PARA OBTENER TIEMPOS DE EJECUCIÓN**

* **omp\_get\_wtime()**
* **omp\_get\_wtick()**