

Variable de control	Ámbito	Valor inicial	¿Qué controla?	Variable de entorno	Rutina para consultar	Rutina para modificar
dyn-var	entorno de datos	true/false (depende de la implementación)	Ajuste dinámico del nº de threads	OMP_DYNAMIC	omp_get_dynamic()	omp_set_dynamic()
nthreads-var	entorno de datos	número (depende de la implementación)	threads en la siguiente ejecución paralela	OMP_NUM_THREADS	omp_get_max_threads()	omp_set_num_threads()
thread-limit-var	entorno de datos	número (depende de la implementación)	Máximo nº de threads para todo	OMP_THREAD_LIMIT	omp_get_thread_limit()	*****NO*****
nest-var		true/false	Paralelismo anidado	OMP_NESTED	omp_get_nested()	omp_set_nested()
run-sched-var	entorno de datos	(kind[,chuk]) (depende de la implementación)	Planificación de bucles para runtime	OMP_SCHEDULE	omp_get_schedule(&kind, &chunk)	omp_set_schedule(&kind, &chunk)
def-sched-var	dispositivo	(kind[,chuk]) (depende de la implementación)	Planificación de bucles por defecto	*****NO*****	*****NO*****	*****NO*****

Otras rutinas del entorno de ejecución:

Nota: void omp_set_schedule (omp_sched_t kind, int chunk_size)

omp_get_thread_num(): devuelve al thread su identificador dentro del grupo de thread.

omp_get_num_threads(): obtiene el número de threads que se están usando en una región paralela. Devuelve 1 en código secuencial.

omp_get_num_procs(): devuelve el número de procesadores disponibles para el programa en el momento de la ejecución.

omp_in_parallel(): devuelve true si se llama a la rutina dentro de una región parallel activa y false en caso contrario.