***Directivas OpenMP***

El preprocesador del compilador las sustituye por código.

#pragma omp es necesario ponerlo en todas las directivas C/C++. El nombre define y controla la acción que se realiza (parallel, for, section…) y las cláusulas especifican adicionalmente la acción o comportamiento.

Las directivas no se tendrán en cuenta si no se compila usando OpenMP.

* Parallel

|  |
| --- |
| #pragma omp parallel *[clause[[,]clause]...]*  *bloque estructurado* |

* Especifica qué cálculos se ejecutarán en paralelo.
* El thread master (el 0) crea un conjunto de threads cuando alcanza dicha directiva.
* Cada thread ejecuta el código incluido en la región que accede al bloque estructurado.
* No reparte tareas entre threads.
* Se puede usar de forma anidada.
* Se usan tantos threads como indique la variable de entorno OMP\_NUm\_THREADS; o bien el fijado por defecto en la implementación (normalmente el número de cpu de un nodo).
* **Directivas para trabajo compartido**
* For

|  |
| --- |
| #pragma omp parallel *[clause[[,]clause]...] bloque estructurado* |

* Paralelismo de datos o paralelismo a nivel de bucle.
* Se tiene que conocer el número de iteraciones, la variable de iteración debe ser un entero. Los bucles no pueden ser de tipo do…while. Además, las iteraciones se deben poder paralelizar.
* La herramienta de programación decide cómo hacer la asignación a no ser que se use la cláusula Schedule.
* Sections

|  |
| --- |
| #pragma omp sections *[clause[[,]clause]...]*  {  [#pragma omp section ]  *structured block*  [#pragma omp section  *structured block* ]  . . .  } |

* La asignación de tareas a threads concretos no la hace el programador, si no la herramienta.
* El número de threads que ejecutan trabajo dentro de un sections coincide con el número de section.
* Single

|  |
| --- |
| #pragma omp single *[clause[[,]clause]...]*  *structured block* |

* Ejecución de un trozo secuencial por un thread.
* Cualquier de los threads puede ejecutar el trabajo del bloque estructurado (no lo controla el programador).

* **Directivas parallel y de trabajo compartido combinadas**

|  |
| --- |
| #pragma omp parallel for *[clauses]*  *for-loop* |

|  |
| --- |
| #pragma omp parallel *[clauses]*  #pragma omp for *[clauses]*  *for-loop* |

|  |
| --- |
| #pragma omp parallel[clauses]  #pragma omp sections *[clauses]*  {  [#pragma omp section ]  *structured block*  [#pragma omp section  *structured block* ]  … } |

|  |
| --- |
| #pragma omp parallel sections *[clauses]* {  [#pragma omp section ]  *structured block*  [#pragma omp section  *structured block* ]  . . .  } |

Las diferencias con la alternativa que no combina son la legibilidad y las prestaciones.

* **Directivas básicas para comunicación y sincronización**
* Barrier
* Barrera: punto en el código en el que los threads se esperan entre sí.
* Al final de parallel y de las construcciones de trabajo compartido hay una barrera implícita.
* Critical

|  |
| --- |
| #pragma omp critical *[(name)] bloque estructurado* |

* Evita que varios threads accedan a variables compartidas a la vez.
* Master

|  |
| --- |
| #pragma omp master *structured block* |

* No es una directiva de trabajo compartido y no tiene barreras implícitas.