

Tarea 1: Estudio del API OpenMP

1.1 OMP DIRECTIVAS:

Una directiva de OpenMP se aplica a un conjunto específico de declaraciones o sentencias compuestas, conocido como región paralela, que tiene una entrada única en la parte superior y una salida única en la parte inferior.

- **#pragma omp parallel:** Esta directiva forma un equipo de hebras e inicia la ejecución paralela. Esta presenta algunas cláusulas:
 - if(scalar-expression)
 - num_threads(integer-expression)
 - default(shared | none)
 - private(list)
 - firstprivate(list)
 - shared(list)
 - copyin(list)
 - reduction(operator: list)

```
#pragma omp parallel [clause[ [, ]clause] ...] new-line  
structured-block
```

- **#pragma omp for:** Esta directiva, en la creación del bucle, se especifica que las iteraciones se distribuirán y ejecutarán por un equipo de hebras. Esta presenta algunas cláusulas:
 - private(list)
 - firstprivate(list)
 - lastprivate(list)
 - reduction(operator: list)
 - schedule(kind[, chunk_size])
 - collapse(n)
 - ordered
 - nowait

```
#pragma omp for [clause[[,] clause] ... ] new-line  
for-loops
```

- **#pragma omp sections:** Esta directiva tiene un conjunto de bloques que tienen que ser distribuidos y ejecutados por un equipo de hebras. Esta presenta algunas cláusulas:
 - private(list)
 - firstprivate(list)
 - lastprivate(list)

- reduction(operator: list)
- nowait

```
#pragma omp sections [clause[[,] clause] ...] new-line
{
    [#pragma omp section new-line]
    structured-block
    [#pragma omp section new-line]
    structured-block ]
...
}
```

- **#pragma omp single:** Esta directiva especifica que el bloque asociado ha de ser ejecutado por una sola hebra del equipo (no necesariamente la hebra master), en el marco de su función implícita. Esta presenta algunas cláusulas:
 - private(list)
 - firstprivate(list)
 - copyprivate(list)
 - nowait

```
#pragma omp single [clause[[,] clause] ...] new-line
structured-block
```

- **#pragma omp task:** Esta directiva define explícitamente una tarea. Esta presenta algunas cláusulas:
 - if(scalar-expression)
 - untied
 - default(shared | none)
 - private(list)
 - firstprivate(list)
 - shared(list)

```
#pragma omp task [clause[ [, ]clause] ...] new-line
structured-block
```

- **#pragma omp master:** Esta directiva especifica que es ejecutada por la hebra principal del equipo (master). No implica sincronización ni a la entrada ni a la salida.

```
#pragma omp master new-line
structured-block
```

- **#pragma omp critical:** Esta directiva restringe la ejecución asociada al bloque por una sola hebra al mismo tiempo.

```
#pragma omp critical [(name)] new-line  
structured-block
```

- **#pragma omp barrier:** Esta directiva especifica de forma explícita un punto de sincronización para todas las hebras.

```
#pragma omp barrier new-line
```

- **#pragma omp taskwait:** Esta directiva especifica que esperar a la finalización de tareas secundarias generadas desde el inicio de la tarea actual.

```
#pragma omp taskwait newline
```

- **#pragma omp atomic:** Esta directiva asegura que un lugar de almacenamiento específico se actualiza automáticamente, exponiéndose a la posibilidad de múltiples y simultáneas escrituras.

```
#pragma omp atomic new-line  
expression-stmt  
  
expression-stmt: one of the following forms:  
x binop = expr  
x++  
++x  
x--  
--x
```

- **#pragma omp threadprivate:** Esta directiva especifica las variables se replican, teniendo cada hebra una copia.

```
#pragma omp threadprivate(list) new-line
```

- **#pragma omp ordered:** Esta directiva especifica que un bloque dentro de una región de bucle se ejecutará en el orden especificado por las iteraciones del bucle.

```
#pragma omp ordered new-line  
structured-block
```

1.2 OMP CLAUSULAS:

CLÁUSULAS DE COMPARTICIÓN DE DATOS:

- **default(shared|none):** Controla el atributo por defecto de compartición de datos para variables referenciadas en el constructor de la región paralela o tarea.

- **shared(list)**: Declara uno o más elementos que se comparten por las tareas generadas por el constructor de la región paralela o tarea.
- **private(list)**: Declara uno o más elementos privados para la tarea.
- **firstprivate(list)**: Declara uno o más elementos privados para la tarea, e inicializados cada uno a su correspondiente valor del elemento original en el momento que se encuentra el constructor.
- **lastprivate(list)**: Declara uno o más elementos privados implícitos a la tarea, y causa que el elemento original se actualiza después de terminar la región.
- **reduction(operator:list)**: Declara una acumulación a partir de una lista de elementos usando un operador asociativo indicado.
- **nowait**: Evita la sincronización implícita de las hebras al terminar el bloque de una directiva.

CLÁUSULAS DE COPIA DE DATOS:

- **copyin(list)**: Copia los valores de las variables privadas de la hebra master en cada una de las otras pertenecientes al equipo de ejecución de la región paralela.
- **copyprivate(list)**: Transmite el valor de un entorno de datos de una tarea a otro entorno de datos de las tareas pertenecientes a la región paralela

1.3 OMP_SCHEDULE:

- **Static**: Las iteraciones son divididas en partes de tamaño indicado, y las partes se asignan a las hebras mediante round-robin en el orden del número de hebras.
- **Dynamic**: Cada hebra ejecuta una parte de las iteraciones, a continuación pide otra parte hasta que no quedan trozos para su distribución.
- **Guided**: Cada hebra ejecuta una parte de las iteraciones, a continuación pide otra parte hasta que no quedan trozos para su distribución. Las partes comienzan siendo grandes y se van reduciendo conforme a lo planificado.
- **Auto**: La decisión de la planificación del reparto es delegada en el compilador o el sistema.
- **Runtime**: El planificador y el tamaño de las partes las toma de run-sched-var (VIC).

1.4 OMP_ENTORNO:

- **OMP_SCHEDULE type[,chunk]**: Establece run-sched-var (VIC) al tipo de planificador y tamaño de las partes. Los tipos válidos para el planificador OpenMP son static, dynamic, guid o auto. Las partes son un entero positivo.
- **OMP_NUM_THREADS num**: Establece nthreads-var (VIC) el número de hebras que usa una región paralela.

- **OMP_DYNAMIC dynamic:** Establece dyn-var (VIC) el ajuste dinámico de las hebras que usa la región paralela. Los valores válidos para dynamic son true o false.
- **OMP_NESTED nested:** Establece nest-var (VIC) a habilitado o deshabilitado el anidamiento paralelo. Los valores válidos para nested es true or false.
- **OMP_STACKSIZE size:** Establece stacksize-var (VIC) que especifica el tamaño de la pila de hebras que crea la implementación de OpenMP. Los valores para size (entero positivo) son size, sizeB, sizeK, sizeM, sizeG. Si la unidad B, K, M o G no se especifica, el tamaño se mide en kilobytes (K).
- **OMP_WAIT_POLICY policy:** Establece wait-policy-var (VIC) que controla el comportamiento esperado de la espera de las hebras. Los valores válidos para policy es activo (las hebras consumen ciclos de procesamiento mientras esperan) y pasivo.
- **OMP_MAX_ACTIVE_LEVELS levels:** Establece max-active-levels-var (VIC) que controla el máximo número de anidamientos activos en una región paralela.
- **OMP_THREAD_LIMIT limit:** Establece thread-limit-var (VIC) que controla el máximo número de hebras que participan en un programa OpenMP.