

Tarea 5. OpenMP

Orientaciones sobre la entrega de la tarea

Realice los ejercicios que aparecen a continuación, cada uno como un proyecto diferente, y suba la respuesta a GitHub o Bitbucket. Comparta en Canvas la URL del repositorio donde aparecen las respuestas a los ejercicios.

Para evaluar la tarea, el profesor compilará y ejecutará el proyecto correspondiente a cada ejercicio de cada estudiante para comprobar que no contienen errores de compilación ni de ejecución, y además, se evaluará el uso de las técnicas correctas de programación vistas en las clases.

La tarea estará activa en la plataforma hasta el 12 de noviembre de 2015 a las 23:55 horas.

No se aceptan trabajos fuera de fecha ni por correo, en ambos casos la calificación de la tarea será 0 puntos.

Pueden auxiliarse de la documentación oficial que aparece en <http://openmp.org/wp/openmp-specifications/> pero los ejemplos que programen no pueden ser iguales a los que aparecen en la documentación oficial de OpenMP.

Utilice la siguiente nomenclatura para el nombre de cada proyecto: omp_directiva. Por ejemplo:

- omp_default
- omp_parallel_for
- omp_simd
- omp_set_num_threads (para funciones use el mismo nombre de la función)
- OMP_SCHEDULE (para variables use el mismo nombre de la variable)

Pablo Zebadúa

Implemente un programa por cada una de las cláusulas y directivas siguientes, donde se muestre el funcionamiento de cada una de ellas:

- `omp parallel`
- `default`
- `private`
- `shared`
- `lastprivate`
- `firstprivate`
- `copyin`
- `copyprivate`
- `reduction`
- `declare reduction`
- `if`
- `num_threads`
- `proc_bind`

José Carlos

Implemente un programa por cada una de las cláusulas y directivas siguientes, donde se muestre el funcionamiento de cada una de ellas:

- `omp sections`
- `omp parallel sections`
- `private`
- `firstprivate`

- lastprivate
- reduction
- nowait

Eric

Implemente un programa por cada una de las construcciones siguientes, donde se muestre el funcionamiento de cada una de ellas, incluyendo el uso de sus diferentes cláusulas:

- omp for
- omp parallel for
- private
- firstprivate
- lastprivate
- reduction
- schedule
- collapse
- ordered
- nowait
- teams
- teams distribute

Juan Andrés Carpio

Implemente un programa por cada una de las construcciones siguientes, donde se muestre el funcionamiento de cada una de ellas, incluyendo el uso de sus diferentes cláusulas:

- omp simd
- safelen

- linear
- aligned
- private
- lastprivate
- reduction
- collapse
- omp declare simd
- simdlen
- uniform
- inbranch
- notinbranch
- omp for simd

César

Implemente un programa por cada una de las construcciones siguientes, donde se muestre el funcionamiento de cada una de ellas, incluyendo el uso de sus diferentes cláusulas:

- target
- target data
- device
- if
- map
- target update
- to
- from
- declare target

- target teams

Jaime Margolin

Implemente un programa por cada una de las construcciones siguientes, donde se muestre el funcionamiento de cada una de ellas, incluyendo el uso de sus diferentes cláusulas:

- teams
- num_teams
- thread_limit
- default
- private
- firstprivate
- shared
- reduction

Pablo Cárdenas

Implemente un programa por cada una de las construcciones siguientes, donde se muestre el funcionamiento de cada una de ellas, incluyendo el uso de sus diferentes cláusulas:

- task
- taskyield
- taskwait
- taskgroup
- cancel
- cancellation point
- flush
- ordered

- if
- final
- untied
- default
- mergeable
- private
- firstprivate
- shared
- depend

Rafael

Implemente un programa por cada una de las construcciones siguientes, donde se muestre el funcionamiento de cada una de ellas, incluyendo el uso de sus diferentes cláusulas:

- single
- barrier
- master
- critical
- atomic
- atomic capture
- flush
- ordered
- cancel
- cancellation point

Rodrigo

Implemente un programa por cada una de las funciones siguientes, donde se muestre el funcionamiento de cada una de ellas:

- `omp_set_num_threads`
- `omp_get_num_threads`
- `omp_get_max_threads`
- `omp_get_thread_num`
- `omp_get_num_procs`
- `omp_in_parallel`
- `omp_set_dynamic`
- `omp_get_dynamic`
- `omp_get_cancellation`
- `omp_set_nested`
- `omp_get_nested`

Alejandra Garay

Implemente un programa por cada una de las funciones siguientes, donde se muestre el funcionamiento de cada una de ellas:

- `omp_set_schedule`
- `omp_get_schedule`
- `omp_get_thread_limit`
- `omp_set_max_active_levels`
- `omp_get_max_active_levels`
- `omp_get_level`
- `omp_get_wtime`
- `omp_get_wtick`

Josué

Implemente un programa por cada una de las funciones siguientes, donde se muestre el funcionamiento de cada una de ellas:

- `omp_get_ancestor_thread_num`
- `omp_get_team_size`
- `omp_get_active_level`
- `omp_in_final`
- `omp_get_proc_bind`
- `omp_set_default_device`
- `omp_get_default_device`
- `omp_get_num_devices`
- `omp_get_num_teams`
- `omp_get_team_num`
- `omp_is_initial_device`

Mauricio Radin

Implemente un programa por cada una de las funciones siguientes, donde se muestre el funcionamiento de cada una de ellas:

- `omp_init_lock`
- `omp_init_nest_lock`
- `omp_destroy_lock`
- `omp_destroy_nest_lock`
- `omp_set_lock`
- `omp_set_nest_lock`
- `omp_unset_lock`
- `omp_unset_nest_lock`
- `omp_test_lock`

- `omp_test_nest_lock`

José Luis

Implemente un programa por cada una de las variables de entorno siguientes, donde se muestre el funcionamiento de cada una de ellas:

- `OMP_SCHEDULE`
- `OMP_NUM_THREADS`
- `OMP_DYNAMIC`
- `OMP_PROC_BIND`
- `OMP_PLACES`
- `OMP_NESTED`
- `OMP_STACKSIZE`
- `OMP_WAIT_POLICY`
- `OMP_MAX_ACTIVE_LEVELS`
- `OMP_THREAD_LIMIT`
- `OMP_CANCELLATION`
- `OMP_DISPLAY_ENV`
- `OMP_DEFAULT_DEVICE`