

# Práctica de OpenMP

Programación Paralela y Distribuida

Curso: 2016/2017

## Índice

1. Normas para la realización de los ejercicios	2
2. Partes a entregar	2
3. Actividad 1	2
4. Actividad 2	2
5. Actividad 3	3
6. Evaluación	3
7. FECHA LIMITE DE ENTREGA	3

Este documento indica los requisitos para resolver la práctica de OpenMP.

## 1. Normas para la realización de los ejercicios

- Realización individual.
- Los trabajos deberán ser entregados obligatoriamente antes de la fecha de entrega fijada en la actividad habilitada en el campus virtual.
- Materiales disponibles: Descargar los recursos para los ejercicios en el campus virtual.
- Recuerde que es OBLIGATORIO entregar los ejercicios para poder evaluar la práctica de OpenMP.
- Cualquier detección de copia de otro alumno, de este o de cursos anteriores, implicará que la práctica está SUSPENSA.

## 2. Partes a entregar

1. Fichero 1: Debe entregarse el código fuente de la práctica, en un fichero .zip que debe llamarse `sopadeletras_` seguido de las iniciales del alumno.
2. Fichero 2: un pdf llamado `Analisis_` seguido de las iniciales del alumno con lo pedido en la actividad
3. Deberán subirse los dos por separado en el campus virtual. Si no se sigue esta nomenclatura la calificación de la práctica será 0 y no será recuperable.

## 3. Actividad 1

Implementar un programa en C o en C++ que resuelva una sopa de letras secuencialmente. La sopa de letras será de tamaño  $N \times N$  y el programa debe funcionar para dicho tamaño  $N$ . Para la prueba inicial se puede usar una sopa de letras estática de al menos  $20 \times 20$  letras. Las palabras a buscar deben ser al menos 10. Quien quiera, para simplificar el problema, puede limitar que las palabras sólo estén en horizontal de izquierda a derecha y en vertical de arriba a abajo. Comprobar que funciona correctamente.

## 4. Actividad 2

Paralelizar el programa anterior con OpenMP y varios procesos, de acuerdo al paradigma maestro/esclavo o trabajadores replicados con bolsa de tareas. El maestro debe ser el que obtenga toda la información de dónde se ha encontrado cada palabra (coordenada de inicio y dirección).

IMPORTANTE: Al ejecutarse el programa debe mostrarse la sopa de letras y las palabras buscadas y en qué ubicación y dirección se han encontrado.

Nota: En el campus virtual tienen un ejemplo de uso de bolsa de tareas con el problema de las N-reinas. No es necesario seguir el mismo procedimiento, pero puede servir de orientación.

## 5. Actividad 3

Comparar los tiempos de ejecución en secuencial y en paralelo; aumentar el tamaño de la sopa de letras hasta ver cómo el programa paralelo es más eficiente. Comentar los resultados.

## 6. Evaluación

- La entrega de la práctica se considerará correcta si funciona correctamente y la explicación es adecuada.
- Las dimensiones a valorar son las siguientes:
  - Corrección de la solución: la práctica funciona correctamente, no hay errores de compilación ni ejecución. Se valorará el encontrar palabras en otras direcciones.
  - Eficiencia de la solución: la solución es eficiente. Se valorará muy favorablemente, el uso (voluntario) distintos modelos de paralelización y su comparativa.
  - Claridad y precisión de las explicaciones: las decisiones de diseño han sido correctamente empleadas. El análisis paralelo vs secuencial está bien realizado (distintas mediciones, número de procesadores y tamaños del problema).
  - Presentación de la memoria: la memoria está correctamente escrita, sin faltas y bien argumentada.

La tabla [1](#) muestra la rúbrica a utilizar para la corrección.

## 7. FECHA LIMITE DE ENTREGA

La dispuesta en el campus virtual.

Concepto/ Factor de multiplicación	0	0,5	1
Corrección (2 puntos)	La práctica no compila o da error de ejecución.	La práctica se ejecuta correctamente, pero hay condiciones de carrera que afectan a la ejecución o no se muestra la salida correctamente.	La práctica funciona correctamente y está bien paralelizada.
Eficiencia (3 puntos)	No hay ningún tipo de paralelismo.	Existe algún tipo de paralelismo, pero no es el más eficiente.	El código está correctamente paralelizado.
Explicaciones (3 puntos)	No se han realizado comparativas ni se ha explicado la solución. El código no está comentado.	Se han realizado algunas comparativas de tiempo o tamaño del problema pero las explicaciones no están bien argumentadas, o la memoria es escueta.	Se han realizado distintas comparativas del tiempo de ejecución con distintos procesadores y tamaños del problema, y las explicaciones están bien argumentadas.
Presentación (2 puntos)	La memoria tiene faltas de ortografía y está mal redactada.	La memoria está correctamente escrita, pero sucinta.	La memoria está bien argumentada y presentada.

Cuadro 1: Rúbrica de la práctica. Nota: hace falta llegar al 50 % de los dos primeros puntos para optar a los dos siguientes.