

CC10A sección 6

Tarea 8

Profesor Mariano Latorre A

Profesores auxiliares: Julio Quinteros, Carlos Reveco, Juan Francisco Rodríguez

Fecha de entrega: 25 de Octubre

Dado que usted es experto en lo que se refiere al trabajo en paralelo usando *threads*, se le pide que implemente los siguientes métodos aprovechando los beneficios de eficiencia que se obtienen usando más de 1 procesador. Note que en todos los métodos usted debe indicar cuantos procesadores dispone para así aprovecharlos al máximo.

- **static public String divisores(long numerogrande, int nprocesadores)**

Se debe dividir el número *numerogrande* (**muy grande**) en *nprocesadores* partes iguales, verificar todos los divisores de esta parte por medio de fuerza bruta (probando todos los números) y entregar todos los números en un *String* separados por un espacio y en orden creciente. Note que *numerogrande* siempre es divisor de *numerogrande*, por lo que siempre se incluye en los resultados. No pruebe con números mayores a *numerogrande/2* para ahorrar cálculos inútiles.

Ej: Tarea8.divisores(20, 2)

Thread1 calcula divisores del 20 entre los números 1 y 5

Thread2 calcula divisores del 20 entre los números 6 y 10

Salida: "1 2 4 5 10 20"

- **static public String multiplos(int n, long numerogrande, int nprocesadores)**

Debe entregar todos los *numerogrande* múltiplos de *n* (desde *n* hasta *n*numerogrande*, donde *numerogrande* es realmente un número grande). Se debe dividir el número *numerogrande* en *nprocesadores* con partes iguales y luego calcular los múltiplos en cada tramo y entregar todos los números en un *String* separados por un espacio y en orden creciente.

Ej: Tarea8.multiplos(2, 10, 2)

Thread1 calcula múltiplos de 2 entre los números 1 y 5

Thread2 calcula múltiplos de 2 entre los números 6 y 10

Salida: "2 4 6 8 10 12 14 16 18 20"

- **static public boolean primo(long numerogrande, int nprocesadores)**

Este método indica si el número *numerogrande* es un número primo. Para esto, se debe dividir el número *numerogrande* (**muy grande**) en *nprocesadores* partes iguales y verificar todos los divisores por medio de fuerza bruta (probando todos los números hasta *numerogrande/2*). Cada thread debe terminar de inmediato si encuentra un divisor que no sea 1. Si el número es primo, debe retornar *true*. Si no lo es, retorna *false*.

Ej: Tarea8.primo(20, 2)

Thread1 calcula divisores del 20 probando con los números entre 1 y 5

Thread2 calcula divisores del 20 probando con los números entre 6 y 10

Salida: false

- **static public double sumatoria(double[] arreglolargo, int nprocesadores)**

Se debe dividir el arreglo *arreglolargo* (**muy largo**) en *nprocesadores* partes iguales, hacer las sumas parciales en cada thread y luego juntar los resultados parciales, retornando el resultado.

Ej: double[] x = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

Tarea8.sumatoria(x, 2)

Thread1 suma números del 1 al 5 = 15

Thread2 suma números del 6 al 10 = 40

Salida: 55

- **static public double menor(double[] arreglolargo, int nprocesadores)**

Se debe dividir el arreglo *arreglolargo* (**muy largo**) en *nprocesadores* partes iguales, se selecciona el mínimo local para cada parte y luego se saca el mínimo global entre los mínimos locales, retornando el resultado.

Ej: double[] x = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

Tarea8.menor(x, 2)

Thread1 mínimo entre 1, 2, 3, 4, 5 = 1

Thread2 mínimo entre 6, 7, 8, 9, 10 = 6

Salida: 1

- **static public double serie(Funcion f, int inicio, int final, int nprocesadores)**

Se debe calcular el resultado de la sumatoria de la *Funcion f* dividiendo el intervalo *[inicio, final]* en *nprocesadores* tramos iguales, y finalmente entregar el resultado final juntando los resultados parciales.

Ej: Funcion f = new Inversa(); //retorna 1/x
Tarea8.serie(f, 1, 20, 2)

$$\text{Thread1: } \sum_{x=1}^{10} \frac{1}{x}$$

$$\text{Thread2: } \sum_{x=11}^{20} \frac{1}{x}$$

Salida: 3.597739565714 (En JAVA puede que el resultado no sea exactamente éste)

Recuerde que la interface *Funcion* exige implementar el método **double valor (double x)** el cual usted deberá utilizar.

Ejemplo:

```
class Seno implements Funcion{
    public Seno(){}
    public double valor(double x){return Math.sin(x);}
}
interface Funcion{
    public double evaluar(double x);
}
```

Importante:

- Todos estos métodos deben estar en la **class Tarea8** en el archivo **Tarea8.java**
- Puede utilizar todas las clases adicionales que necesite.
- **Debe** respetar los nombres de los **métodos y clases** incluyendo las mayúsculas.
- Recuerde incluir el encabezado con sus datos.
- Recuerde que el proceso main siempre es un Thread (procesador) y no puede quedar de ocioso mientras los otros trabajan.
- Cada método vale 1 punto.
- Habrá un ejemplo en ucursos de un par de funciones con su respectivo método valor.
- Los ejemplos son sólo demostrativos y para la corrección se utilizarán números largos o arreglos con muchos números.
- No se puede usar la clase consola.
- No se aceptan atrasos.
- Consultas sólo en U-Cursos y en el foro “Consultas de Tareas”.