



UNIVERSIDAD
DE ALMERÍA

ESTRUCTURAS DE DATOS Y ALGORITMOS II

Introducción a los hilos (Threads)

GEDAS - Grupo de Trabajo de las EDAs (EDA I y EDA II)

ÍNDICE

- Conveniencia
- Implementaciones
- Heredar de Thread
- Implementar Runnable

1

Threads
Conveniencia



UNIVERSIDAD
DE ALMERÍA

Introducción a los hilos. Conveniencia.

- La microelectrónica optó hace unos años por incorporar más núcleos de procesamiento en lugar de aumentar la frecuencia.
- La mayoría de los microprocesadores actuales tienen varios núcleos que, a su vez, pueden incorporar más de un hilo lógico en los mismos. Los Snapdragon de los Smartphone suelen tener 8.
- Los algoritmos clásicos se diseñaron para CPU,s de un sólo núcleo.
- Ejecutar un algoritmo de orden de ejecución elevado en un solo hilo, desaprovecha el hardware disponible.
- Algunos de los algoritmos que vamos a estudiar son fácilmente “paralelizables” sin mucho trabajo extra de programación.

2

Threads
Implementaciones

Introducción a los hilos, Implementaciones.

- En java existen varios métodos básicos para hacer funcionar un programa en paralelo.
 - Heredar de Thread. Hacemos que la clase que va a ser paralelizada tenga como padre la clase Thread.
 - Implementar Runnable. Implementando la interfaz Runnable en la clase que se vaya a paralelizar.
 - El método `Runtime.getRuntime().availableProcessors()` devuelve el número de procesadores disponibles en la máquina virtual Java.

Introducción a los hilos, la clase Thread.

- La clase Thread está diseñada para facilitar la programación concurrente.
- Tiene métodos para poder solucionar tareas clásicas como el productor-consumidor.
- Para nuestros propósitos, los métodos que nos interesan son:
 - `public void run()`
 - `public void start()`
 - `public final void join() throws InterruptedException`

Introducción a los hilos, la clase Thread.

- Método run()
 - Debe ser implementado anulando el original (override).
 - El código que queramos ejecutar ha de ser incluido en este método.
 - Puede llamar a otros métodos de la clase, que se ejecutarán en el hilo creado al lanzar la instancia.
 - Una vez finalizado el método, el hilo pasará al estado “muerto” y dejará los recursos hardware para otras tareas.

Introducción a los hilos, la clase Thread.

- Método start()
 - No hay que implementarlo, se usa el original de la clase.
 - Para lanzar una instancia de nuestra clase, necesitamos usar este método:
 - `MiClase.start();`
 - Una vez lanzada la tarea, el programa continua con la siguiente instrucción, no espera a la finalización de la misma.
 - Hay distintos métodos para conocer el estado de las tareas paralelas que no son objeto de esta introducción.

Introducción a los hilos, la clase Thread.

- Método join()
 - Espera a que el hilo termine.
 - Puede terminar por la finalización del método run(), por causas externas o por métodos propios para tal efecto.
 - Este método para la ejecución del hilo que lo llama hasta que se termine la ejecución de la instancia.
 - Si tenemos varios hilos, se puede esperar secuencialmente, pero también hay otras posibilidades.

Introducción a los hilos, la clase Thread.

- Ejemplo con un programa DyV para lanzar dos hilos
 - class QuickSortThread extends Thread{
 - private int[] vector;
 - private int bajo, alto;
 - public QuickSortThread(int[] v, int bajo, int alto) {
 - super();
 - this.vector = v;
 - this.bajo = bajo;
 - this.alto = alto;
 - }
 - @Override
 - public void run() {
 - this.quickSort(this.bajo, this.alto);
 - }

Introducción a los hilos, la clase Thread.

- Continuación
 - `private void quickSort(int bajo, int alto) {`
 - `if(bajo < alto) {`
 - `int p = pivote(bajo, alto);`
 - `quickSort(bajo, p - 1);`
 - `quickSort(p + 1, alto);`
 - `}`
 - `}`
 - `public int pivote(int bajo, int alto){ ... }`
 - `***` (otros métodos conocidos o triviales)
 - `//Fin de la clase.`

Introducción a los hilos, la clase Thread.

- Llamada a una ejecución paralela con dos hilos:
 - `Int[] v = inicializaVector(Cantidad);`
 - `QuickSortThread qst = new QuickSortThread(v, 0, cantidad - 1);`
 - `int p = qst.pivote(0, Cantidad - 1);`
 - `QuickSortThread qst1 = new QuickSortThread(v, 0, p - 1);`
 - `QuickSortThread qst2 = new QuickSortThread(v, p, Cantidad - 1);`
 - `qst1.start();`
 - `qst2.start();`
 - `qst1.join();`
 - `qst2.join();`

Introducción a los hilos, interfaz Runnable.

- Java no permite herencia múltiple.
- Para poder utilizar hilos en estos casos, podemos implementar la interfaz Runnable.
- Al implementarla debemos escribir, anulando (override) el original, el método `public void run()`.
- Para usar esta variante lo hacemos, como anteriormente, de la siguiente forma:
 - `MiClaseRunnable mcr = new MiClaseRunnable(...);`
 - `Thread t = new Thread(mcr);`
 - `t.start();`
 - `t.join();`

Introducción a los hilos, Bibliografía.

- <http://informatica.uv.es/iiguia/LP/teoria/apuntes/cuatr1/tema3%20concurrency2.pdf>
- <https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html>



UNIVERSIDAD
DE ALMERÍA

ESTRUCTURAS DE DATOS Y ALGORITMOS II

Introducción a los hilos (Threads)

GEDAS - Grupo de Trabajo de las EDAs (EDA I y EDA II)