

# Computación Paralela

## Departamento de Ciencias de la Computación

### Hilos

Ing. Carlos Andrés Pillajo B, Msc.  
capillajo@espe.edu.ec

28 de noviembre de 2023

1 Introducción

2 Hilos en Java

3 Referencias

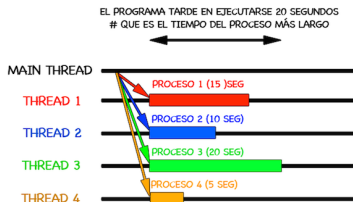
## 1 Introducción

## 2 Hilos en Java

## 3 Referencias

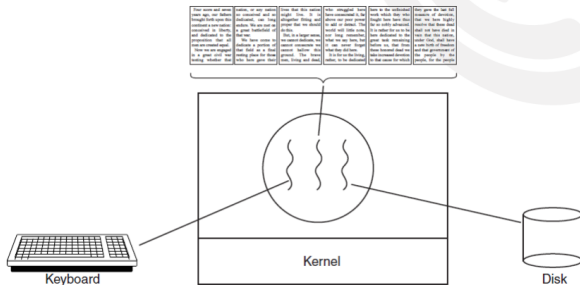
# Introducción

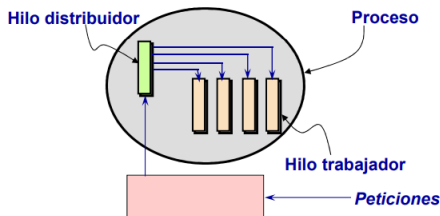
- Los hilos son unidades de ejecución más pequeñas dentro de un proceso que pueden ejecutarse de forma independiente y concurrente.
- Comparten recursos como memoria y archivos, pero tienen su propio contador de programa, registros y pila de ejecución.
- La implementación de hilos permite aprovechar la concurrencia para mejorar el rendimiento de un programa al realizar múltiples tareas de forma simultánea.



# Aplicaciones

- Un navegador web puede tener un hilo para mostrar imágenes o texto, mientras que otro hilo recupera datos de la red.
- Un procesador de texto puede tener un hilo para mostrar gráficos, otro hilo para responder al tecleo del usuario y un tercer hilo para realizar la revisión ortográfica y gramatical en segundo plano.





- Pueden crearse a medida que se necesitan y destruirse al finalizar la tarea encomendada.
- Pueden existir siempre y quedar libres al finalizar la tarea encomendada
  - Más eficiente (evita el trabajo de crear y destruir hilos)

## ■ Ejemplo: Servidor secuencial de ficheros

Algoritmo

```
Mientras no haya que terminar {  
  Esperar a que llegue una petición  
  Comprobar que la petición es correcta  
  Si (los datos no están en la caché) {  
    Realizar operación de E/S bloqueante sobre disco  
  }  
  Enviar resultado  
}
```

Caché para bloques accedidos recientemente

- Sencillo
- Prestaciones pobres: permanece bloqueado

## ■ Ejemplo (cont.): Servidor de ficheros con múltiples hilos

**Hilo distribuidor**

Algoritmo 1

```
Mientras no haya que terminar {  
  Esperar a que llegue una petición  
  Esperar trabajador libre  
}
```

**Hilo trabajador**

Algoritmo 2

```
Mientras no haya que terminar {  
  Esperar trabajo  
  Comprobar que la petición es correcta  
  Si (los datos no están en la caché) {  
    Realizar operación de E/S  
    bloqueante sobre disco  
  }  
  Enviar resultado  
  Avisar que está libre  
}
```

- Mayor complejidad
- Buenas prestaciones

## ● Hilos a Nivel de Usuario (User-Level Threads)

- Los hilos son gestionados por la aplicación sin intervención del sistema operativo.
- La creación, planificación y sincronización de hilos se realizan en el espacio de usuario.
- Tres librerías principales: POSIX Pthreads, Win32 threads, Java threads.

## ● Hilos a Nivel de Kernel (Kernel-Level Threads)

- La administración de hilos se realiza por el sistema operativo a nivel de kernel.
- Ofrece una mayor concurrencia, pero puede tener más sobrecarga.
- Ejemplos: Windows XP/2000/Vista/7/8, Solaris (de Sun, ahora Oracle), Tru64 UNIX (de Digital, luego Compaq, finalmente HP), Mac OS X (Apple).



- **Paralelismo**

- Permite realizar múltiples tareas al mismo tiempo, aprovechando los recursos del sistema.

- **Responsividad**

- Los hilos pueden mejorar la capacidad de respuesta de una aplicación, ya que una tarea puede continuar ejecutándose mientras otras esperan, en lugar de bloquear todo el proceso.

- **Sincronización**

- Es crucial sincronizar los hilos para evitar problemas como condiciones de carrera y lecturas/escrituras conflictivas.

- **Overhead**

- La creación y gestión de hilos pueden tener un costo, y un exceso de hilos puede conducir a una mayor sobrecarga.

- **Paralelismo de Datos**

- Los hilos operan en diferentes conjuntos de datos de manera simultánea.

- **Paralelismo de Tareas**

- Los hilos ejecutan tareas independientes al mismo tiempo.

- Una biblioteca de hilos proporciona al programador una API para crear y administrar hilos.
- Algunos lenguajes de programación proporcionan bibliotecas o extensiones para trabajar con hilos, como pthread en C/C++, Thread en Java, y threading en Python.

# Contenido

1 Introducción

2 Hilos en Java

3 Referencias

- Existen dos formas: crear una nueva clase como subclase de la clase Thread o declarar una clase e implementar la interfaz Runnable.
  - **Uso de la SubClase:** Crear la subclase de Thread y sobrescribir el método run() con el código que se ejecutará por el hilo.
  - **Uso de la interface Runnable:** La clase debe implementar la interface Runnable, desarrollar el método run() con el código que se ejecutará por el hilo.

# Ejecución de un Thread

- Primero se debe instanciar el Thread.
- Ejecutar el método `start()`.
  - Esto es para que la JVM lo pueda ejecutar como un thread realizando la llamada al método `run()`.

# Código Uso de la SubClase

```
1
2  ✓ public class MiThread extends Thread {
3
4      MiThread(String nombreHilo) {
5          super(nombreHilo);
6      }
7
8  ✓ public void run() {
9      for(int i = 0; i < 10; i++)
10         System.out.println(i + " - " + getName());
11         System.out.println("Hilo finalizado");
12     }
13
14  ✓ public static void main(String[] args) {
15
16         new MiThread("Julio").start();
17         new MiThread("Lucas").start();
18
19     }
20
21 }
```



# Código Uso de la interface Runnable

```
1
2  ✓ public class MiRunnable implements Runnable {
3
4  ✓      public void run() {
5              for(int i = 0; i < 10; i++)
6                  System.out.println(i + " - " + Thread.currentThread().getName());
7                  System.out.println("Hilo finalizado");
8
9      }
10
11     public static void main(String[] args) {
12         new Thread(new MiRunnable(), "Julio").start();
13         new Thread(new MiRunnable(), "Lucas").start();
14     }
15
16 }
```

# Contenido

1 Introducción

2 Hilos en Java

3 Referencias



## Parallel Programming

*Wilkinson, B. and Allen, M.*

Second Edition



## Ejercicios de programación paralela con OpenMP y MPI.

*Román, J.E., Alonso, J.M, Alvarruiz, F., Blanquer, I., Guerrero, D.,  
Ibáñez, J.J., Ramos, E.*

Universitat Politècnica de Valencia