



Centro de Enseñanza Técnica Industrial

Desarrollo de Software

Estableciendo prioridades en la ejecución de hilos

Jesús Alberto Aréchiga Carrillo

22310439 5N

Profesor

José Francisco Pérez Reyes

Octubre 2024

Guadalajara, Jalisco

Introducción

En esta práctica se utiliza el código de la práctica anterior para implementar prioridades en los hilos, el objetivo es manejar las prioridades para que un hilo se ejecute primero y ya que haya acabado, se ejecute el otro hilo.

Desarrollo

Código para implementar:

```
public class ThreadExample implements Runnable{
    public void run(){
        for(int i = 0; i < 10 ; i++)
            System.out.println(i + " " +
                Thread.currentThread().getName());

        System.out.println("Termina thread " +
            Thread.currentThread().getName());
    }

    public static void main(String [] args){
        Thread pepe = new Thread(new ThreadExample(), "Pepe");
        Thread juan = new Thread(new ThreadExample(), "Juan");

        pepe.setPriority(10);
        juan.setPriority(1);

        pepe.start();
        juan.start();

        System.out.println("Termina thread main");
    }
}
```

Resultados:

Windows:

GNU/Linux:

Termina thread main	Termina thread main
0 Juan	0 Juan
0 Pepe	1 Juan
1 Pepe	2 Juan
2 Pepe	3 Juan
3 Pepe	4 Juan
4 Pepe	5 Juan
1 Juan	6 Juan
5 Pepe	7 Juan
2 Juan	8 Juan
6 Pepe	9 Juan
3 Juan	0 Pepe
7 Pepe	1 Pepe
4 Juan	2 Pepe
8 Pepe	3 Pepe
5 Juan	4 Pepe
9 Pepe	5 Pepe
6 Juan	6 Pepe
7 Juan	7 Pepe
8 Juan	8 Pepe
9 Juan	9 Pepe
Termina thread Pepe	Termina thread Pepe
Termina thread Juan	Termina thread Juan

Implementando el código en Windows no parece haber una diferencia notable con respecto al objetivo. En cambio, cuando se ejecuta en GNU/Linux, si hace el funcionamiento como debe. Leyendo un poco en la documentación, se puede encontrar que la prioridad de los hilos realmente depende del planificador de hilos del sistema operativo que lo esté implementando.

Conclusiones

Sabemos que el planificador de hilos de Windows tiende a ser apropiativo y basado en tiempos. Es decir, asigna intervalos de tiempo a cada hilo y puede cambiar de un hilo a otro sin importar su prioridad, con el objetivo de garantizar que todos los hilos tengan una oportunidad de ejecutarse. La prioridad puede influir, pero Windows decide no detener completamente los hilos de menor prioridad y sigue repartiendo el tiempo entre todos los hilos. Por lo tanto, no siempre se respeta estrictamente el orden de ejecución basado en la prioridad.

En Linux, el planificador de hilos tiende a ser más estricto en cuanto a la prioridad de los hilos. Si se asigna una prioridad más alta a un hilo (especialmente usando `Thread.MAX_PRIORITY`), el planificador le da más tiempo de CPU, o incluso ejecuta por completo ese hilo antes de pasar a hilos con menor prioridad. Linux sigue un modelo de planificación más orientado a las prioridades.