

UNIDAD 2 – PROGRAMACIÓN MULTITHILO (RESUMEN)

La unidad trata el concepto de hilos (threads), elementos fundamentales para la programación concurrente. Un hilo es una unidad ligera de ejecución dentro de un proceso, capaz de ejecutarse de forma independiente pero compartiendo memoria y recursos con otros hilos del mismo proceso.

1. CONCEPTO DE HILO Y ESTADOS

Un hilo tiene varios estados: nuevo, preparado, en ejecución, bloqueado, dormido, en espera y terminado. El paso entre estados se produce al llamar métodos como `start()`, `sleep()`, `wait()`, `notify()`, `join()` o por decisiones del planificador.

2. MODELOS DE HILOS

Se diferencia entre hilos de usuario y de sistema. También se explican modelos de relación entre hilos y kernel: muchos a uno, uno a uno y muchos a muchos.

3. CREACIÓN DE HILOS EN JAVA

Java permite crear hilos de dos formas:

- Extender la clase `Thread`.
- Implementar la interfaz `Runnable`.

En ambos casos debe sobrescribirse el método `run()`, donde va el código a ejecutar en paralelo. La ejecución real del hilo comienza con `start()`, nunca con `run()` directamente.

4. SINCRONIZACIÓN Y RECURSOS COMPARTIDOS

La concurrencia provoca problemas cuando varios hilos acceden a recursos compartidos. Java aporta mecanismos para evitar inconsistencias:

- `synchronized` para proteger secciones críticas.
- `wait()`, `notify()` y `notifyAll()` para comunicación entre hilos.
- Semáforos, monitores y colas como `BlockingQueue`.
- `volatile` para evitar optimizaciones erróneas y garantizar visibilidad entre hilos.

5. PROBLEMAS TÍPICOS EN MULTITHILO

Se explican problemas como:

- Interbloqueo (deadlock): dos hilos se bloquean mutuamente esperando recursos.
- Inanición: un hilo nunca obtiene CPU por baja prioridad.

La solución suele basarse en mejorar el orden de adquisición de bloqueos y usar prioridades adecuadamente.

6. PRIORIDADES Y DEMONIOS

Los hilos pueden configurarse con prioridades (`MAX_PRIORITY`, `NORM_PRIORITY`, `MIN_PRIORITY`).

Un hilo demonio es un hilo en segundo plano que finaliza cuando termina el último hilo de usuario.

7. GRUPOS DE HILOS Y TAREAS PERIÓDICAS

ThreadGroup permite organizar grupos de hilos para administrarlos conjuntamente.

Java también incluye Timer y TimerTask (o javax.swing.Timer) para programar tareas periódicas.

En resumen, la unidad presenta las bases teóricas y prácticas para trabajar con hilos en Java, resolver problemas de concurrencia y aplicar mecanismos de sincronización que garanticen la correcta ejecución de aplicaciones multihilo.