



Programación de Servicios y Procesos

3.5 Anexo I - Debugging aplicaciones multihilo en Netbeans



I.E.S.
Doctor Balmis

Apuntes de PSP (https://psp2dam.github.io/psp_sources/es/) creados por Vicente Martínez bajo licencia

CC BY-NC-SA 4.0  (<http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>)

3.5 Anexo I - Debugging aplicaciones multihilo en Netbeans

Los entornos de desarrollo Java proporcionan las herramientas necesarias para la depuración (debugging) de aplicaciones multihilo.

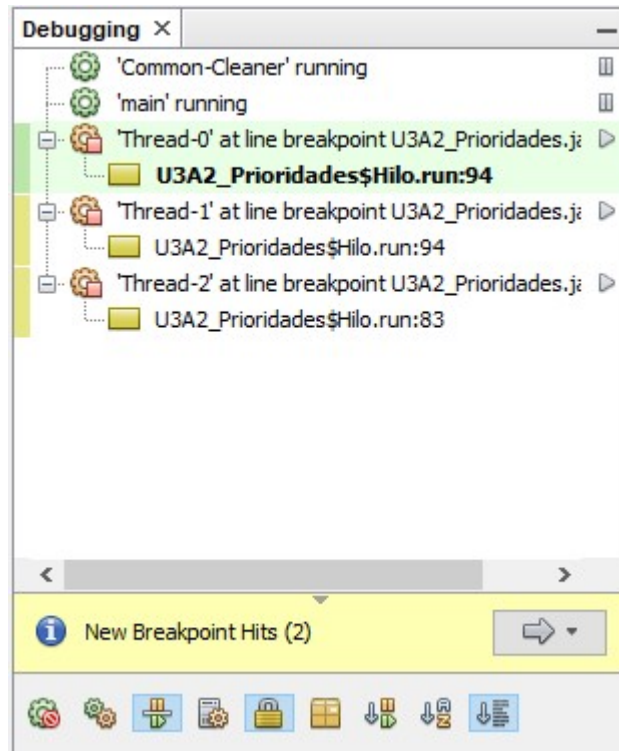
Al igual que en aplicaciones monohilo, podemos ubicar los breakpoints en el punto en el que nos interese detener la ejecución del programa, para de esta forma poder inspeccionar el estado de los objetos, los valores de las propiedades, etc.

Para poder monitorizar los hilos de un programa, debemos activar la ventana de Debugging desde el menu *Window > Debugging > Debugging (Alt+Shift +9)*

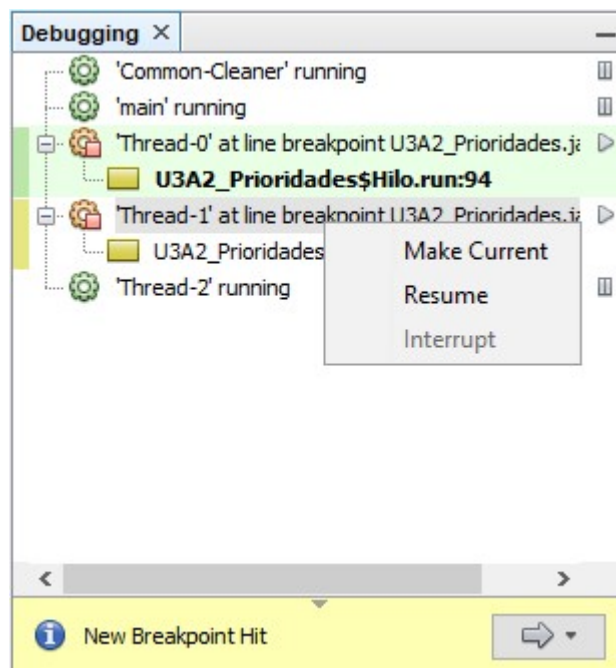


A partir de ese momento podremos ver la parte derecha las opciones de depuración de hilos junto con toda la información de los hilos que se están ejecutando en nuestra aplicación.

Veamos el significado de lo que Netbeans nos muestra en esta caja de diálogo



- Con una franja de color verde a la izquierda y resaltado también en color verde vemos lo que Netbeans denomina el *current thread*, es decir, el hilo que estamos depurando y sobre el que se aplicarán las acciones de depuración como Step Into, StepOver, Pause, Continue, etc. Además, podremos inspeccionar el valor de las variables para ese hilo en concreto.
- Con una franja de color amarillo en la parte izquierda, y relacionado con el aviso de la parte inferior, vemos los hilos que están parados en algún breakpoint marcado por nosotros. Este color y los avisos están para que atendamos al hilo y podamos visualizar su estado o continuar con su ejecución.
- En la parte derecha de cada hilo tenemos un acceso rápido a la función de Reanudar o Pausar cada uno de los threads activos.



Pulsando con el botón derecho sobre cualquiera de los hilos podemos convertirlo en el *current thread*. Así pasamos a controlar la ejecución e inspección de ese hilo mientras los demás siguen pausados o bien continúan su ejecución.



Además, la rueda dentada que está al lado del identificador de cada hilo nos da información del estado del hilo.

- Cuando la rueda está de color naranja indica que el hilo está suspendido en espera de nuestra atención.
- Cuando la rueda está verde, indica que el hilo está en ejecución. Si ponemos el ratón encima podemos obtener información de su estado. Como se puede observar en la imagen, el hilo 2 está en ejecución aunque se encuentra en estado de espera. Es un hilo que está bloqueado en un join.



Por último, la depuración de hilos también nos ayuda a detectar cómo los hilos están sincronizándose con el uso de monitores. Podemos ver si un hilo tiene un monitor (bloqueo) bajo su control, y qué hilos están esperando a que se libere un bloqueo para poder continuar.

Como herramienta adicional, Netbeans nos proporciona una utilidad (*Debug > Check for deadlocks*) que comprueba si hay hilos que estén en un interbloqueo, informándonos de los monitores que tienen bajo su control y de los monitores para los que están realizando alguna espera.