

z2-Environment - Bug #2161

Application Threadpool gets confused on interruption

02.01.2024 14:22 - Henning Blohm

Status:	In Progress	Start date:	02.01.2024
Priority:	Normal	Due date:	
Assignee:	Henning Blohm	% Done:	0%
Category:	z2-core	Estimated time:	0.00 hour
Target version:	2.10.1		
origin:			

Description

When interrupting a thread in the thread pool, it will terminate but still be in the pool. This happens in WorkerThread.java:

```
        if (this.wm.taskCompleted(this)) {
            synchronized (this) {
                if (!this.kicked) {
                    // only if not having been kicked between being put into
                    // the pool and entering this block
                    waiting = true;
                    try {
                        this.wait();
                    } catch (InterruptedException e) {
                        logger.fine("Thread interrupted: "+this);
                    } finally {
                        this.waiting = false;
                    }
                } else
                    this.kicked = false;
            }
        } else
            synchronized (this) {} // also to comply to the memory model
```

The code is super outdated and far to complex.

Acceptance Criteria

- The ThreadPool code is simplified with less concurrency spaghetti code
- When a thread is interrupted, it is terminating and remove from the pool
- The fix is merged into master

History

#1 - 02.01.2024 14:22 - Henning Blohm

- Target version changed from 2.11 to 2.9.5

#2 - 02.01.2024 14:24 - Henning Blohm

- Description updated

#3 - 07.01.2024 16:49 - Henning Blohm

- Target version changed from 2.9.5 to 2.11

#4 - 23.01.2024 11:39 - Henning Blohm

- Status changed from New to In Progress

#5 - 25.04.2024 16:37 - Henning Blohm

- Target version changed from 2.11 to 2.10.1