Tread Pool Implementation

Four programs showing the way a Thread Pool works.

The program includes the generation of 20 tasks and their insertion in a FIFO queue.

Then a set of workersextract the tasks from the queue and process them.

Service.java is the main program. You can compile it through the following command:

javac -cp ..\lib\commons-cli-1.4\commons-cli-1.4.jar;. Service.java

The running configuration can be controlled through the following command line options:

-i time between consecutive task insertions in the queue

-d task duration

-p number of workers (pool size)

The program can be activated through the following command:

java -cp ..\lib\commons-cli-1.4\commons-cli-1.4.jar;. Service -p <…> -i <…> -d <…>

The program prints the following results:

Time: < from start> : Enqueueing Task n. <Task Number>

Time: < from start>: Task n. <Task Number> in Worker <Thread – Id> Starting

Time: < from start>: Task n. <Task Number> in Worker <Thread – Id> Done

Here is a set of interesting command line options to be tested:

1. 20 Tasks on 10 Workers, each Task lasts 1000 msec. Tasks are generated every 500 msecs

-p 10 -i 500 -d 1000

2. 20 Tasks on 10 Workers, each Task lasts 2000 msec. Tasks are generated every 100 msecs

-p 10 -i 100 -d 2000

3. 20 Tasks on 1 Worker, each Task lasts 1000 msec. Tasks are generated every 100 msecs

-p 1 -i 100 -d 1000