Marcel ZAUDER 16-124-836
Pascal GERIG 16-104-721

8.1 Several Questions

a) *Why are servers (e.g., web servers) usually structured as thread-per-message gateways?*

b) *What are condition objects?*

c) *Why does the SimpleConditionObject from the lecture not need any instance variables?*

d) *What are "permits" and "latches"?*

8.2 Questions about Futures

a) *Which implementation would you prefer for this kind of problem? Is there any considerable difference at all? Justify your answer!*

b) *Write a new class FutureTaskExecDemo.java that uses an ExecutorService implementation to compute the future task and to execute the clients, instead of creating explicit new threads. What is the benefit of using executors?*

c) *Add a time constraint such that the client thread waits for at most a given amount of time for the result.*

8.3 Nested Monitor

In order to avoid the deadlock, the *synchronized* keywords need to be removed from the *put()* and *get()* method in the *TheNest* buffer. Otherwise only one of each thread can either enter the *put()* or *get()* method, therefore the farmer blocking the access for the hen to put the egg into the nest, but because the farmer is waiting for the hen to notify the farmer that an egg was laid into the buffer, we have a deadlock.

8.4 Thread Speed Evaluation

a) *What amount of processing cores does the CPU in your notebook have and what's the model / manufacturer of it?*
Intel Core i7-9700k, 8-Core Processor

b) *Does the implementation scale well, i.e., more concurrent threads help greatly to reduce the overall calculation time? Please provide concrete runtimes you experienced!*

Total Amount of Threads	Time [ms]	Result (until first error)
10	3	3.0...
50	8	3.12...
100	11	3.13...
500	37	3.13...
1000	74	3.140...
5000	319	3.1413...
10000	666	3.1414...
50000	2896	3.14157...
100000	5577	3.14158...
500000	25992	3.141590...
1000000	54899	3.141591...

c) *Depending on your results, why or why not does the solution scale well?*

d) *How would you improve the runtime with respect to faster calculations (without changing the algorithm)?*

e) *Which algorithm would you recommend as drop-in replacement for the Leibniz formula for faster calculation?*

f) *Why do the runtimes with identical parameters vary so much?*