Lab Session 07

Home exercises

1. **[5p]** Implement producer consumer with semaphores and a buffer of size 5.
2. **[5p]** Read and run the code from the support with an implementation of a lock.
    - Not to be used in real life.

Lab Exercises

1. **[10p]** Solve multipleProducersMultipleConsumers using ArrayBlockingQueue.
2. **[20p]** Solve the synchronizationProblem using AtomicInteger.
3. **[20p]** Solve the bug in bugConcurrentHashMap using methods from ConcurrentHashMap.
4. **[20p]** Write a program with four threads.
    - Three of the threads read elements from the files elements1.txt, elements2.txt and elements 3.txt and puts them in a list.
    - Thread number four sorts the elements from the list.
    - The list can only be sorted after all elements are inserted.
    - Use a Semaphore to block thread four until the list if full.
5. **[20p]** Write a program with three threads starting from parallelTree.
    - Two thread reads the name of the nodes of a tree followed by the name of the parent from the files.
    - Thread number three needs to verify that the tree is correctly constructed.
    - Thread number three should wait on a CyclicBarrier until the other threads are done building the tree.
    - You cannot make use of a global lock. You need to use a lock for each node in the tree.
    - The files are created so that child nodes always appear after their parents. However, a parent can be in a different file from the child.