BLG546E - Object Oriented Concurrent Programming

10.04.2013 - MIDTERM I

Duration: 120 min.

1) (30p)

a) (15p) Below is a randomly ordered list of data structures, parallelization concepts and techniques used in implementation. Match each element in the columns correspondingly and explain your answer with a few sentences.

Data Structure	**Technique**	**Concept**
Linked list parallelization	Elimination Array	Blocking Parallelization
Queue parallelization	Hand-over-hand Locking	Fine grained Locking
Stack parallelization	Monitor Locks	Non-blocking Parallelization

b) (15p)

Explain the difference between the following terms:

- Wait-free parallelization
- Lock-free parallelization
- Starvation-free parallelization

2) (30p) An Anderson Lock is a Queue based lock which can be implemented on arrays. A simple explanation on how the Anderson Lock works is given below.

The threads share a tail field, initially zero. To acquire the lock, each thread atomically increments tail. Call the resulting value the thread's slot. The slot is used as an index into a Boolean flag array. If flag[j] is true, then the thread with slot j has permission to acquire the lock. Initially, flag[0] is true. To acquire the lock, a thread spins until the flag at its slot becomes true. To release the lock, the thread sets the flag at its slot to false, and sets the flag at the next slot to true. All arithmetic is modulo n, where n is at least as large as the maximum number of concurrent threads. Please keep in mind that each thread locally keeps its slot index.

a) (20p) Implement the Anderson Lock using Java.
b) (5p) How does Anderson lock avoid contention?
c) (5p) How does false sharing effect Anderson lock?

3) (30 p) For the following two sub-questions you are going to use a trivial custom random number generator. Your random number generator is going to produce the following numbers in a cyclic manner:

3 - 5 - 2 - 1 - 4

a) (10p) Place the following numbers in a skip list.

20 - 40 - 90 - 30 - 60 - 70 - 50 - 80 - 10

b) (30p) 5 threads are performing operations on a lazily parallelized skip list which contains the numbers in (a). Given the timing diagram below, please illustrate the status of the list at the end of each marked checkpoint time slot in the diagram. Please note

- the threads may claim a lock any time during their execution

- threads release claimed locks at the last time unit of their execution

- contains(45) returns true

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
add(45)	▨	▨	▨	▨	▨	▨	▨			
remove(45)					▨	▨	▨	▨	▨	▨
add(47)		▨	▨	▨						
contains(45)							▨	▨		
remove(40)		▨	▨	▨	▨	▨				
Checkpoints					(I)		(II)	(III)	(IV)	

In your answer please clearly indicate the locked nodes and the owner of the lock.