Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

Integrated Computer Science
Year 4 Annual Examinations

CS4021 Advanced Computer Architecture

4 January 2017  Goldsmith Hall  09.30– 11.30

Dr Jeremy Jones

Instructions to Candidates:

Answer THREE questions.
All questions carry equal marks
All questions are marked out of 20
Q1. Briefly describe the operation of the Peterson lock.

```c
// Peterson lock
int flag[2];  // initially 0
int last;    // initially 0

void acquire(int id) {  // id is the thread ID [0 or 1]
    int j = 1 - id;
    flag[id] = 1;
    last = id;
    while (flag[j] && last == id);
}

void release(int id) {  // id is the thread ID [0 or 1]
    flag[id] = 0;
}
```

[4 marks]

In terms of the Peterson lock, describe the four desirable properties of a concurrent algorithm.

[4 marks]

What is Promela? What is Spin? How can Spin be used to check the operation of a concurrent algorithm? What is Linear Temporal Logic (LTL)?

[4 marks]

Translate the Peterson lock into Promela code and show how Spin can be used to test whether the algorithm has the four desirable properties?

[8 marks]
Q2. What is an atomic instruction? Briefly describe the atomic instructions supported by the IA32/x64 instruction set.

[4 marks]

Provide sample C/C++ code and explain its operation for (i) a testAndTestAndSet lock (ii) a ticket lock and (iii) an array lock.

[8 marks]

In terms of lock performance, explain why the underlying cache coherency protocol needs to be taken into consideration. What is the advantage of an array lock compared with a ticket lock?

[4 marks]

Explain what happens if a thread is pre-empted while holding a ticket lock when there are more threads than CPUs.

[4 marks]
Q3 What is a lockless data structure? What is the advantage of using a lockless data structure compared with one protected by a lock?

[1 marks]

Consider the implementation of a lockless ordered linked list. Nodes are normally removed using a two-step process (1) marking the node and (2) removing the node from the list by updating a pointer. Explain why this is necessary.

[4 marks]

Consider the following pseudo code for the find function in the implementation of a lockless ordered linked list.

```c
find(head **NodeType; key KeyType) : Boolean {
  retry:
    pred <- head;
    cur <- *pred;
    while (curr != NULL) {
      next <- curr^.next;
      if (next & 1) {
        if !CAS((volatile PVOID*) prev, cur, next-1) != curr)
          goto retry;
      }
      else {
        ckey <- curr^.key;
        if (*prev != cur)
          goto retry;
        if (ckey >= key)
          return (ckey == key);
      }
    }
  return 0;
}
```

(Question 3 continues on next page)
Briefly explain the operation of the above code.

[4 marks]

Write companion functions for adding and removing items from the lockless linked list.

[4 marks]

What is the ABA problem? Discuss the relevance of the ABA problem if the above code reused retired nodes? Discuss how hazard pointers can be used to solve this problem. Illustrate your answer by outlining the code that would need to be added to support hazard pointers.

[7 marks]
Q4. What is a transaction in terms of transactional memory? What is hardware lock elision (HLE)? What are its advantages? Explain the operation of the Intel TSX instructions that support HLE. Explain, in detail, how the first level cache is used to detect conflicts.

[8 marks]

Given code which updates a binary search tree (BST) protected by a lock, show how the locking code can be modified to support HLE. Explain how the code works, particularly for the cases when the lock is taken with and without elision. Show how the code can be structured to increase the probability of obtaining the lock with elision.

[8 marks]

Comment on (1) expected speed up compared with using RTM (restricted transactional memory) directly and (2) why calls to memory management functions (e.g. malloc and free) should or should not be part of a transaction.

[4 marks]