8 Concurrent and Distributed Systems (RNW)

The developers of *FictionalOS* have an implementation of the *Network File System version 3 (NFSv3)* without support for distributed locking. As a result, applications experience race conditions when operating concurrently on files in NFS. Rather than using the hideously complex NFS distributed locking protocol, the OS developers decide to develop their own simpler locking mechanism ("How hard can it be?"). They add two new NFS *remote procedure calls* (RPCs) that can be used on file nodes in NFS: `NFS_LOCK` and `NFS_UNLOCK`, which lock and unlock nodes, respectively. These are implemented through simple atomic operations on the in-memory `node` data structure representing a file on the NFS file server:

```
nfs_lock(node) {
  atomic {
    if (node->lock_held != 0)
      return (FAILURE);
    node->lock_held = 1;
    return (SUCCESS);
  }
}
```

```
nfs_unlock(node) {
  atomic {
    node->lock_held = 0;
    return (SUCCESS);
  }
}
```

SunRPC retransmission ensures reliable delivery when packets are lost. If a client receives `FAILURE`, it will issue new RPCs each second until it receives `SUCCESS`.

(*a*) Explain why, when server reboots, concurrent applications writing to files across multiple nodes may suffer data races despite acquiring suitable locks, and describe a solution to this problem. [4 marks]

(*b*) (*i*) Define *at-least once* RPC semantics. [1 mark]

(*ii*) Explain why *at-least once* RPC semantics may cause trouble for each of the `NFS_LOCK` and `NFS_UNLOCK` RPCs. [4 marks]

(*c*) Explain why the polling nature of the `NFS_LOCK` RPC, as described, makes it difficult to implement reliable server-side deadlock detection. [4 marks]

(*d*) (*i*) Define *priority inheritance* and explain what problem it solves. [2 marks]

(*ii*) Describe the changes to the `NFS_LOCK` and `NFS_UNLOCK` RPCs necessary to implement priority inheritance. Include any new RPC arguments and return values. Explain the changes (and limitations) this imposes on software implementations. [5 marks]