Politecnico di Milano – V Facoltà di Ingegneria

088924 – Tecnologie di middleware per sist. dist. (Ord. 270)

Prof. G. Cugola - January 27, 2011

Rules:

- **You are not allowed to use books, notes, or other material.**
- **You can answer in Italian or English.**
- **Total time for the test: 1.5 hour.**

---

1. Implement a `ReadWriteLock` class in Java. The class offers three methods for locking: `readLock` to acquire the lock in read (non exclusive) mode, `writeLock` to acquire the lock in write (exclusive) mode, and `unlock`, to release the lock. More than one thread can hold the read lock simultaneously. At any given time, only one thread can hold the write lock, which can be granted only if no other thread is holding a read or write lock. A thread calling a `readLock` or `writeLock` when the lock cannot be granted is suspended until the lock is freed.

2. Describe the main differences between RMI and Corba as two middleware systems offering a RPC-like form of communication.

3. Suppose you have to implement a videogame organized in rounds (like a table game, e.g., monopoly). Imagine the different participants have to share some state (e.g, who owns what, where is each participant on the table, etc.). Which middleware would you prefer among those described during the course? Why? Provide also an architectural sketch of the solution (use UML, fragments of code and any other notation you think may help your description).

4. Consider the following TinyOS interfaces:

a la solución puede ayudar la pag 5 de 12-wsn-resumen

```
interface Foo {
  command uint8_t op1(uint8_t data);
  event void event1Fired(uint16_t data);
  command result_t op2();
  event void op2Done();
}

interface Boo {
  command void op3(uint8_t data);
  event void op3Done();
}
```

Write the skeleton (empty operations) of a module that *uses* interface `Foo` and *provides* interface `Boo`.