Politecnico di Milano – V Facoltà di Ingegneria

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

Prof. G. Cugola – September 21, 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 `SyncStrIntMap` class in Java. It takes a `Map<String, Integer>` (supposed to be non thread safe) at creation time and implements a thread save version of the map. It provides the usual `get`, `put`, `remove`, `containsKey`, and `size` methods. The first three methods suspend the caller if the passed key has not an associated value (methods `get` and `remove`) or if it has already one (method `put`).

2. Write the IDL interface of a CORBA server to monitor temperature and humidity in a building. You should be able to remotely read either the temperature or the humidity of a single room (identified by a number) and to register a listener to be notified when the temperature in any room exceeds a given threshold (passed as a parameter). Write all the interfaces and data structures that must be public for the system to operate.

3. Write in TinyOS the interface of a module to read data (8 bit integer numbers) from an ADC converter. The ADC may operate in single shot mode or periodically. The former mode uses a split phase operation to read single data values. The latter mode is enabled through a `start` operation that specifies the period and a `stop` operation to interrupt reading.

4. a) Write a higher order "map" function that applies a given function to all the elements of a list. Also show the code for calling it.
   b) Write a modified version of the previous function that demands the computation for each element to a remote server, waiting for the result before moving to the next element.

   The server is on the node 'server@serverhost', running the following start function:

```
loop() -> receive {SenderNode, Fun, Num} ->
      {client, SenderNode} ! Fun(Num),
      loop()
end.

start() ->
 Pid = spawn(fun() -> loop() end),
 register(server, Pid).
```