

IS: Esercizio 30/1/2013

Si consideri un sistema in cui un numero non noto a priori di Thread devono accedere ad una risorsa in mutua esclusione (non piu' di un Thread all'interno della zona critica).

Si utilizzi la classe AtomicInteger per realizzare una classe Lock con i metodi e la semantica seguente:

- 1) Public boolean tryLock()
 - a. Il metodo torna true se e solo se e' riuscito ad acquisire il lock, false altrimenti, senza attese;
- 2) Public boolean unlock()
 - a. Il metodo rilascia il lock e permette accessi ulteriori

Non e' richiesta la rientranza del metodo tryLock.

tryLock ritorna immediatamente con il responso.

Punti in discussione

E' necessario garantire che non possa succedere che due Thread possano "acquisire il lock", e' possibile garantire che un'acquisizione non sia mai ritardata inutilmente?