

Programmazione I

Prova di Programmazione – 25 giugno 2015 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma di gestione dell'accesso ad una risorsa condivisa. Ciascuna entità che compete per l'accesso alla risorsa è individuata da un numero intero **id**. All'inizio la risorsa è *libera*, ossia la prima entità che la vuole utilizzare ottiene immediatamente l'accesso alla risorsa, la quale passa a sua volta in stato *occupato*, e vi rimane finché non viene liberata esplicitamente. Se una entità prova ad accedere alla risorsa mentre la risorsa è occupata, tale entità viene inserita in fondo alla coda delle entità in attesa della risorsa. Chiamiamo Q tale coda. Se la coda Q non è vuota quando la risorsa è liberata, allora la risorsa viene occupata dall'entità in testa alla coda. Altrimenti la risorsa passa in stato libero. La lunghezza massima della coda Q è definita a tempo di scrittura del programma. Il programma deve fornire le seguenti funzionalità.

1. **prendi_risorsa(id)** Fa sì che l'entità **id** occupi la risorsa, se è libera. Se invece la risorsa è occupata, allora inserisce l'entità in fondo alla coda Q . Una stessa entità può invocare questa funzionalità anche mentre già occupa la risorsa, o mentre è già accodato in attesa della risorsa.
2. **stampa_stato** Stampa lo stato della risorsa, e, se la risorsa è occupata, stampa da quale *id* è occupata, più gli *id* delle eventuali entità accodate. Ad esempio, se la risorsa è occupata dall'entità 3, e ci sono le entità 5, 3 e 7 accodate, allora stampa:
Occupata da 3
In coda: 5 3 7
3. **salva_stato** Salva su un file binario lo stato della risorsa e l'eventuale lista di entità accodate.
4. **carica_stato** Carica lo stato della risorsa e l'eventuale lista di entità accodate. Il precedente stato è perso.
5. **libera_risorsa** Libera la risorsa. In particolare, la risorsa torna nello stato libero se non c'è nessuna entità nella coda Q , altrimenti viene occupata dall'entità in testa alla coda.
6. **prendi_risorsa_2(id)** Identico a **prendi_risorsa**, ma dando priorità all'entità **id**. In particolare, se la risorsa è occupata e solo se la seguente condizione è vera, allora l'entità **id** viene inserita in testa, e non in fondo, alla coda Q . La condizione è che nessuna delle eventuali entità già accodate sia rimasta accodata *per troppo tempo*. In merito, misuriamo il tempo in numero di volte in la risorsa è liberata, e poniamo il limite a 2, ossia, per un'entità accodata è passato troppo tempo se l'entità è rimasta continuamente accodata mentre la risorsa veniva liberata per almeno due volte.

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'inserimento di dati in formato errato e di messaggi troppo lunghi da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale o di materiale didattico di altra natura
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1 e 2. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati