

**Prova pratica del 21/02/2014**  
**Durata della prova: 150 minuti**

*Cognome ..... Nome ..... Matr. ....*

Lo studente legga attentamente il testo e produca il programma, il makefile, ed i casi di test necessari per dimostrarne il funzionamento. La mancata compilazione dell'elaborato, la compilazione con errori o l'esecuzione errata del programma daranno luogo alla valutazione come **prova non superata**. Ricordarsi di indicare Nome, Cognome e matricola su questo stesso foglio, che dovrà essere in ogni caso consegnato alla Commissione. Al termine della prova lo studente dovrà fare verificare il funzionamento del programma ad un membro della Commissione.

**Testo della prova**

Si realizzi in linguaggio C o C++ un programma multi-thread (con la libreria **PThreads**) che simuli il monitoraggio di traffico ferroviario. Il programma dovrà essere basato sul costrutto **Monitor**.

Si supponga di monitorare 4 treni, la cui posizione (la stazione in cui si trova il treno) sia rappresentata da un valore intero compreso tra 0 e 10. La posizione del treno viene aggiornata da dei thread **capitreno**, e consultata da dei thread denominati **viaggiatori**. Ciascun treno deve essere monitorato usando una istanza distinta del monitor così definito:

```
struct monitor_treno {  
    // riporta la stazione in cui si trova il treno (intero tra 0 e 10)  
    int stazione;  
  
    // aggiungere qui variabili per la sincronizzazione  
    ...  
};  
  
void inizializza(struct monitor_treno * m);  
void rimuovi(struct monitor_treno * m);  
int leggi_stazione(struct monitor_treno * m);  
void scrivi_stazione(struct monitor_treno * m, int stazione);
```

Il metodo **leggi\_stazione()** dovrà restituire la posizione attuale del treno, permettendo a più viaggiatori di leggere in contemporanea. Il metodo **scrivi\_stazione()** dovrà permettere ai capitreno di aggiornare la posizione del treno, garantendo la mutua esclusione tra thread.

Il programma principale dovrà istanziare 4 istanze del monitor e 4 thread capitreno (una istanza e un thread per ogni treno). I capitreno dovranno invocare per 10 volte il metodo **scrivi\_stazione()**, modificando il valore della posizione ad ogni invocazione (incrementando il valore di 1) e attendendo 3 secondi tra le invocazioni. Il valore della stazione deve essere inizialmente posto a 0.

Inoltre, dovranno essere istanziati 10 thread viaggiatori. I thread viaggiatori dovranno scegliere un treno a caso, e dovranno consultare la posizione del treno scelto per 3 volte, invocando il metodo **leggi\_stazione()** dopo avere atteso per un tempo casuale (tra 1 e 6 secondi) tra le invocazioni.