

- Non bo a simulare  
il test sulle var. di scope  
per farlo con il while  
e non con l'if

**Prova pratica del 15/05/2014**  
**Durata della prova: 150 minuti**

Cognome BRANCACCIO Nome ALESSANDRO Matr. N46000590

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**) per la gestione delle consegne di una pizzeria. Il programma dovrà essere basato sul costrutto **Monitor**.

```
struct consegna {
    int pizza;
    int stato;
    int cliente;
};
struct Monitor {
    struct consegna c[5]; // vettore di consegne da gestire
    ... // aggiungere qui variabili per la sincronizzazione
};
void aggiungi_consegna(struct Monitor * m, int pizza, int cliente);
int inizia_consegna(struct Monitor * m); // ritorna un indice del vettore "c"
void fine_consegna(struct Monitor * m, int indice_consegna);
```

Il metodo **aggiungi\_consegna()** dovrà essere invocato da un thread **pizzaiolo**, che inserisce una nuova consegna, specificando la pizza (un valore a caso) e l'identificativo del cliente (un numero, da incrementare ogni volta che si aggiunge una consegna). Il metodo cerca un elemento del vettore **c** nello stato **VUOTO** (inizialmente, tutti gli elementi del vettore sono in questo stato), ne modifica i campi **pizza** e **cliente** usando i parametri di ingresso, e pone lo stato dell'elemento a **PRONTO**. Se non vi è almeno un elemento nello stato **VUOTO**, il thread dovrà essere posto in attesa di questa condizione.

Il metodo **inizia\_consegna()** dovrà essere invocato dai thread **fattorino**. Se non vi è almeno un elemento del vettore nello stato **PRONTO**, il metodo dovrà mettere il thread in attesa di questa condizione. Il metodo deve fare una ricerca nel vettore, cercando un elemento nello stato **PRONTO**. Trovato un elemento, il metodo dovrà porre lo stato a **IN\_CONSEGNA**, e restituire in uscita l'indice dell'elemento.

Dopo aver invocato **inizia\_consegna()**, il thread **fattorino** attende un tempo casuale compreso tra 1 e 3 secondi (per simulare la consegna), dopodiché invoca il metodo **fine\_consegna()** per informare che la consegna è stata effettuata. Esso pone l'elemento indicato dal parametro **indice\_consegna** nello stato **VUOTO**.

Il programma dovrà istanziare 3 thread **fattorini**, ciascuno dei quali effettuerà 5 consegne. Inoltre, dovrà istanziare un thread **pizzaiolo** che aggiungerà 15 consegne, attendendo un intervallo di tempo di 1 secondo tra una aggiunta e l'altra.