

Università degli Studi di Napoli Federico II
Corso di Laurea in Ingegneria Informatica
Esame di Sistemi Operativi
Proff. Cotroneo, Natella

Prova pratica del 12/05/2015
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/C++ un'applicazione **multithread** utilizzando il costrutto **Monitor**, che simuli la gestione di traffico aereo. Per gestire l'accesso al vettore di voli (si assume un massimo di 3 voli nello spazio aereo), si utilizzi la soluzione del **produttore/consumatore con vettore di stato**.

```
typedef struct {
    int identificativo;
    int quota;
} Volo;
typedef struct {
    Volo vettore_voli[3];
    int vettore_stato[3];          // LIBERO, OCCUPATO, IN USO
    // ... inserire qui le variabili per la sincronizzazione ...
} GestioneVoli;
void InserisciVolo(GestioneVoli * g, int identificativo);
void RimuoviVolo(GestioneVoli * g, int identificativo);
void AggiornaVolo(GestioneVoli * g, int identificativo, int quota);
```

Il metodo *InserisciVolo* fa da produttore, ed inserisce un volo con quota iniziale 0 e con l'identificativo indicato. L'inserimento dura 1 secondo (durante cui un elemento del vettore è posto "IN USO", per poi divenire "OCCUPATO"), da simulare con la primitiva *sleep*. Il metodo *InserisciVolo* deve **sospendere il thread chiamante se non vi è spazio disponibile** nel vettore.

Il metodo *RimuoviVolo* rimuove dal vettore il volo indicato dal parametro *identificativo*, facendo una ricerca nel vettore per trovarne la posizione. A differenza della soluzione vista al corso, il metodo *RimuoviVolo* **non deve sospendere il thread chiamante** in alcun caso. La rimozione dura 1 secondo (durante cui un elemento del vettore è posto "IN USO", per poi divenire "LIBERO").

Il metodo *AggiornaVolo* deve, nell'ordine: (1) cercare la posizione del volo (indicato da *identificativo*, e il cui stato sia "OCCUPATO"); (2) porre l'elemento "IN USO"; (3) modificare la quota del volo (parametro *quota*); (4) porre l'elemento nello stato "OCCUPATO". *AggiornaVolo* **non deve sospendere il thread chiamante** in alcun caso. L'aggiornamento dura (a caso) tra 1 e 3 secondi.

Il programma principale dovrà creare 5 thread, utilizzando un ciclo, e passando l'indice del ciclo ad ogni thread figlio. Ogni thread dovrà inserire un nuovo volo (usando come identificativo il valore passato dal padre), fare tre aggiornamenti su quel volo (aggiornando la quota a 1000, 2000, e 1000), ed infine rimuoverlo. L'istanza del monitor può essere acceduta (a scelta dello studente) dai thread attraverso una variabile globale, oppure attraverso passaggio di parametri.