

Testo della prova

Si realizzi in linguaggio C/C++ un programma **multiprocesso**, basato sul **costrutto Monitor (di tipo signal-and-wait di Hoare)**, che realizzi lo schema del **produttore-consumatore con vettore di stato**. È previsto che i produttori possano inserire elementi di due tipi differenti. I buffer del vettore possono essere in 4 possibili stati: LIBERO, IN_USO, OCCUPATO1, OCCUPATO2 (questi ultimi due stati a seconda del tipo di elemento che è stato prodotto). I produttori chiamano uno fra due differenti metodi ("produci_tipo_1" e "produci_tipo_2") per effettuare la produzione sui due tipi differenti. Analogamente, i consumatori chiamano uno fra due metodi differenti ("consuma_tipo_1" e "consuma_tipo_2") per consumare.

Per verificare il funzionamento del programma, si creino 2 processi produttori, che effettuino 4 produzioni ciascuno di tipo 1; altri 2 processi produttori, che effettuino 4 produzioni ciascuno di tipo 2; 1 processo consumatore, che effettui 8 consumazioni di tipo 1; 1 processo consumatore, che effettui 8 consumazioni di tipo 1. Il vettore di buffer conterrà al più 4 elementi.

```
#define DIM 4
```

```
typedef struct {
    int vettore[DIM];
    int stato[DIM];
    int num_liberi;
    int num_occupati_tipo1;
    int num_occupati_tipo2;
    // ... aggiungere ulteriori variabili per la sincronizzazione
} MonitorPC;

void inizializza(MonitorPC * p);
void rimuovi(MonitorPC * p);
void produci_tipo_1(MonitorPC * p, int valore);
void produci_tipo_2(MonitorPC * p, int valore);
int consuma_tipo_1(MonitorPC * p);
int consuma_tipo_2(MonitorPC * p);
```