Sistemi Operativi (modulo II / B) CON SOLUZIONE 19 giugno 2019

Esercizio 1

Considerare i seguenti thread che modificano la variabile globale `x`:

```
int x = 10; // variabile globale condivisa da T1 e T2
thread T1 {              thread T2 {
      x = x+1;                 x = x+1;
}                        }
```

1. Mostrare un'esecuzione di T1 e T2 al termine della quale la variabile x vale `11` invece che `12`: Può succedere che il valore 10 venga caricato in un registro contemporaneamente da T1 e T2, venga quindi incrementato nel registro (11) e scritto in memoria. Entrambi i thread scriveranno 11.

2. Si consideri una variabile globale `locked` inizializzata a `false` e discutere perché la soluzione proposta qui sotto per proteggere la *sezione critica* **non** risolve il problema del punto precedente:
   ```
   while(locked){}
   locked=true;
   x = x+1; // sezione critica
   locked=false;
   ```
   Sia T1 che T2 possono superare il while contemporaneamente accedendo alla sezione critica e modificando il valore di x simultaneamente. Questa soluzione non garantisce mutua esclusione.

Esercizio 2

Si consideri una **variante del problema dei filosofi** in cui i filosofi raccolgono **due bacchette qualsiasi** tra le 5 disponibili.

1. Realizzare una soluzione basata su semafori (completare il codice qui sotto, **commentandolo** opportunamente, indicando i valori di inizializzazione dei semafori):
   ```
   sem bacchette = 5; // Ci sono 5 bacchette, inizializziamo il semaforo a 5
   Thread filosofo(i) {
     while(1) {
       // pensa
       P(bacchette);  // raccoglie una bacchetta qualsiasi
       P(bacchette);  // raccoglie una bacchetta qualsiasi
       // mangia
       V(bacchette);  // deposita una bacchetta
       V(bacchette);  // deposita una bacchetta
     }
   }
   ```

2. Discutere la possibilità di stallo e proporre una soluzione, basata su semafori, che lo prevenga:
   Se tutti i filosofi si allocano una bacchetta il semaforo `bacchette` diventerà rosso (`0`) e i 5 filosofi si bloccheranno tutti in attesa della seconda bacchetta. E' una situazione di attesa circolare irrisolvibile, ovvero uno stallo. Lo stallo si può prevenire aggiungendo un semaforo `posti`, inizializzato a 4, che permette solo a 4 filosofi di raccogliere le bacchette, evitando l'attesa circolare.. E' sufficiente allocare un posto prima della raccolta e rilasciarlo dopo il rilascio delle bacchette:
   ```
   // pensa
   P(posti); // alloca un posto dei 4 disponibili
   P(bacchette); P(bacchette); // raccoglie le 2 bacchette
   // mangia
   V(bacchette); V(bacchette); // deposita le 2 bacchette
   V(posti); // rilascia un posto
   ```