

# — COMPITO CON SOLUZIONE —

## Sistemi Operativi (modulo II / B)

10 settembre 2014

### Esercizio 1

Si considerino 2 processi P1, P2 che si sincronizzano tramite i semafori contatori S1, S2, inizializzati a 2 e 0, rispettivamente:

```
process P1 {                                process P2 {
    for (i = 1; i<=3; i++) {                  for (i = 1; i<=3; i++) {
        P(S1);                                P(S2);
        <A>                                    <B>
        V(S2);                                V(S1);
    }                                          }
}
```

Elencare tutte le combinazioni in cui possono essere eseguiti i blocchi di codice A e B. È possibile che alcuni blocchi vengano eseguiti in parallelo (ad esempio su 2 core)? Spiegare brevemente.

Lo schema è quello classico Produttore-Consumatore con un buffer di un due elementi (infatti il semaforo per la produzione S1 è inizializzato a 2). I blocchi vengono eseguiti in uno dei modi seguenti:

A,A,B,B,A,B.

A,A,B,A,B,B.

A,B,A,B,A,B.

A,B,A,A,B,B.

In pratica abbiamo tutte le sequenze in cui, in ogni punto dell'esecuzione, il numero di B è minore uguale al numero di A e il numero di A è minore uguale al numero di B più 2 (buffer di 2 elementi). Quando i due semafori valgono 1 è possibile che A e B vengano eseguiti assieme in quanto nessuno dei due semafori è bloccante. Indicando con il simbolo | questa possibilità otteniamo la sequenza: A,A|B,A|B,B.

**Esercizio 2**

Considerare il problema dei filosofi a cena in cui ogni filosofo raccoglie 2 bacchette scelte a caso:

```
semaphore bacchette[5]={1,1,1,1,1};
int sx,dx;

Filosofo(i) {
    while(1) {
        < pensa >

        sx = random(5); // sceglie un random da 0 a 4
        do {
            dx = random(5); // sceglie un random da 0 a 4
        } while (dx == sx);

        P(bacchette[sx]);
        P(bacchette[dx]);
        < mangia >
        V(bacchette[sx]);
        V(bacchette[dx]);
    }
}
```

Discutere la possibilità di stallo e proporre una soluzione che prevenga tale problema, modificando opportunamente il codice.

Se due filosofi scelgono ad esempio le bacchette 0,1 e 1,0 potrebbe accadere che eseguano la prima P (sui semafori 0 e 1) e si blocchino sulla seconda entrando in attesa circolare irrisolvibile (stallo).

Per prevenire il problema si può fare in modo che le bacchette vengano allocate sempre in ordine crescente, secondo la strategia di allocazione gerarchica:

```
if (sx < dx) {
    P(bacchette[sx]); P(bacchette[dx]);
} else {
    P(bacchette[dx]); P(bacchette[sx]);
}
```

Lo stallo non si verifica più perché diventa impossibile la formazione di attesa circolare.