

# — COMPITO CON SOLUZIONE —

## Sistemi Operativi (modulo II / B)

15 gennaio 2014

### Esercizio 1

Si considerino 2 processi P1, P2 che eseguono il seguente codice:

```
process P1 {                                process P2 {
    print "S";                                print "S";
    print "Y";                                print "T";
    print "E";                                print "M";
}                                              }
```

Sincronizzare P1 e P2 tramite semafori (indicando il valore di inizializzazione di ogni semaforo e aggiungendo le necessarie P e V) in modo che, indipendentemente dallo scheduling, stampino sempre la stringa "SYSTEM". Spiegare brevemente la soluzione proposta.

**SOLUZIONE:** Utilizziamo 3 semafori binari inizializzati a false (rosso) in modo che siano bloccanti. Ogni volta che un processo deve attendere la stampa dell'altro precediamo la print con una P. Quando l'altro processo ha stampato mettiamo la corrispondente V.

```
process P1 {                                process P2 {
    print "S";                                P(s1);
    print "Y";                                print "S";
    V(s1); P(s2);                            print "T";
    print "E";                                V(s2); P(s3);
    V(s3);                                    print "M";
}                                              }
```

### Esercizio 2

Si devono sincronizzare N thread `corridore` in modo che attendano il segnale dello `starter` prima di iniziare la corsa.

```
thread corridore {                            thread starter {
    corsa.attendiStart();                        /* attendi l'orario di partenza */

    /* corri piu' forte che puoi! */            corsa.prontiVia()
}                                              }
```

Scrivere il codice del monitor `corsa`. Il metodo `attendiStart` deve mettere in attesa i thread `corridore` fino a quando verrà invocato (dallo `starter`) il metodo `prontiVia`. Spiegare brevemente.

**SOLUZIONE:** Utilizziamo `wait` e `notifyAll` in quando dobbiamo sbloccare tutti i corridori assieme. In questo modo è sufficiente una condition sola `start` sulla quale tutti i corridori attendono.

```
monitor corsa{
    condition start;

    attendiStart() {
        start.wait;
    }

    prontiVia() {
        start.notifyAll;
    }
}
```