

# — COMPITO CON SOLUZIONE —

## Sistemi Operativi (modulo II / B)

4 giugno 2013

### Esercizio 1

Si considerino 2 processi P1, P2 che si sincronizzano tramite i semafori contatori S1, S2, inizializzati a 1 e 0, rispettivamente:

```
process P1 {                                process P2 {
    for (i = 1; i<=5; i++) {                for (i = 1; i<=5; i++) {
        P(S1);                               P(S2);
        <A>                                   <B>
        V(S2);                               V(S1);
    }                                         }
}
```

In quale ordine vengono eseguiti i blocchi di codice A e B? È possibile che alcuni blocchi vengano eseguiti in parallelo? Spiegare brevemente.

Lo schema è quello classico Produttore-Consumatore con un buffer di un elemento solo (infatti il semaforo per la produzione S1 è inizializzato a 1). I blocchi vengono quindi eseguiti nell'ordine:

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

Inizialmente P2 si blocca sul semaforo (rosso) S2. P1 invece esegue P(S1) e porta il semaforo a 0, esegue A e poi sblocca P2 (oppure rende S2 verde se P2 non è ancora stato eseguito). Se P1 tenta di eseguire nuovamente A si blocca sul semaforo S1 (rosso). Solo quando P2 esegue B e fa la V(S1), P1 può proseguire. Ecco che si crea una alternanza A,B,A,B,... che prosegue fino alla fine del ciclo for.

I blocchi A e B non vengono mai eseguiti in parallelo perché almeno uno tra S1 e S2 sarà rosso bloccando uno dei due processi.

### Esercizio 2

Si considerino 3 processi P1, P2 e P3 che utilizzano 3 file f1, f2 e f3 in scrittura (w). La open in scrittura è bloccante se il file è già in uso (aperto) da un altro processo.

```
process P1 {                                process P2 {                                process P3 {
    open(f1,"w");                            open(f2,"w");                            open(f3,"w");
    open(f2,"w");                            open(f3,"w");                            open(f1,"w");
    ...                                       ...                                       ...
}
```

Discutere la possibilità di stallo e proporre, eventualmente, una soluzione che prevenga tale problema.

Se i tre processi aprono il primo file accade che f1, f2 e f3 sono aperti e le 3 open successive sono bloccanti. Ogni processo attende il processo successivo e si verifica uno stallo (attesa circolare). In particolare P1 attende P2, P2 attende P3 e P3 attende P1.

Per prevenire il problema si può usare una tecnica di allocazione gerarchica: un processo può chiedere un file solo se non ne possiede già uno di livello gerarchico inferiore. Se ordiniamo i file in modo naturale  $f1 < f2 < f3$ , abbiamo che P3 viola tale allocazione e deve necessariamente chiedere prima f1 e poi f3:

```
process P3 {
    open(f1,"w");
    open(f3,"w");
    ...
}
```

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