

semafori-barbiere

- . Un negozio di barbiere ha un barbiere, una poltrona da barbiere e n sedie per i clienti in attesa
- . Se non ci sono clienti, il barbiere si mette sulla sedia da barbiere e si addormenta
- . Quando arriva un cliente, sveglia il barbiere addormentato e si fa tagliare i capelli sulla sedia da barbiere
- . Se arriva un cliente mentre il barbiere sta tagliando i capelli a un altro cliente, il cliente si mette in attesa su una delle sedie
- . Se tutte le sedie sono occupate, il cliente se ne va scocciato!

Soluzione

Possiamo vedere questo problema come un problema di produttore-consumatore. I produttori sono i clienti, che “producono” se stessi. E il consumatore è il barbiere, che consuma un cliente alla volta.

Innanzitutto, dobbiamo descrivere la “vita” del barbiere e dei clienti, perchè non è esattamente specificata:

```
process Barbieri {
    while (true) {
        attendi cliente
        // taglia capelli
        libera poltrona
    }
}

process Cliente {
    verifica se c'è posto, altrimenti esci
    attendi turno
    // fatti tagliare i capelli
}
```

E questo è il programma:

```
Semaphore mutex = new Semaphore(1);
Semaphore clienteDisponibile = new Semaphore(0);
Semaphore poltrona = new Semaphore(0);
int sedieLibere = N;

process Barbieri {
    while (true) {
        clienteDisponibile.P();
        // taglia capelli
        poltrona.V();
    }
}

process Cliente {
    mutex.P();
    if (sedieLibere==0) {
        mutex.V();
        return;
    }
    sedieLibere--;
    mutex.V();
    clienteDisponibile.V();
    poltrona.P();
    mutex.P();
    sedieLibere++;
    mutex.V();
    // Fatti tagliare i capelli
    // e poi esci
}
```

Alcuni errori visti nelle soluzioni degli studenti:

```
while(sedieLibere==0) // Inserito all'inizio del codice cliente
; //non fa niente
```

Questo è busy waiting e non rispetta le richieste del problema (se ne va indignato). Qui in realtà attende, ma senza mettersi in coda sul semaforo.

Utilizzo di un semaforo “sedie” inizializzato a N. Bisogna essere molto cauti nell'utilizzare una soluzione del genere: se il cliente si sospende su “sedie”, i primi N clienti passano (senza in realtà aspettare), mentre i clienti in eccesso attendono (mentre dovrebbero andarsene).