

# Ingegneria del Software: 6/3/2013

---

Il paradigma di computazione **Bulk Synchronous Processes (BSP)** assume che ogni processo di un gruppo possa essere in una delle due seguenti fasi:

- 1) calcolo;
- 2) comunicazione.

Durante la fase di calcolo i processi non possono effettuare alcuna comunicazione, durante la fase di comunicazione i processi possono solo scambiarsi informazioni (messaggi).

Ogni processo puo' dichiarare la sua intenzione di cambiare fase, ma non la cambiera' effettivamente sino a che non siano pronti a farlo (e abbiano dichiarato tale intenzione) anche tutti gli altri.

Scrivere un programma Java in grado di fornire le seguenti primitive di gestione BSP:

**goToProcess()** dichiara l'intenzione del processo chiamante di passare dalla fase comunicazione alla fase processing. Non ritorna sino a che non e' stata chiamata la stessa primitiva anche da tutti gli altri i processi del gruppo;

**goToCom()** come sopra, ma in senso inverso;

**send(ID, msg)** invia un messaggio, non ritorna sino a che il messaggio non e' stato ricevuto;

**recv(ID)** riceve un messaggio, non ritorna sino a che non c'e' un messaggio da ritornare.

L'uso di una primitiva in una fase non coerente deve generare un'eccezione di tipo adeguato (definito dall'utente).