

Variabili condizione

Meccanismo di sincronizzazione che permette di aspettare efficientemente (no busy waiting) un cambiamento a stato condiviso protetto da un lock.

wait rilascia il lock e sospende l'esecuzione del chiamante, inserendolo in una lista associata alla CV (atomicamente). Riacquisisce il lock prima di restituire il controllo.

signal rimuove un thread dalla lista della CV e lo mette in stato *ready*. Deve essere chiamato in una sezione protetta dal lock.

broadcast signal su tutti i thread della lista.

Sono senza memoria.

Utilizzo

<code>lock.acquire();</code>	<code>lock.acquire();</code>
<code>while(!cond)</code>	<code>...</code>
<code> cv.wait(lock);</code>	<code>cv.signal();</code>
<code>...</code>	<code>lock.release();</code>
<code>lock.release();</code>	

È fondamentale che il rilascio del lock avvenga atomicamente (cioè nella sezione protetta dallo stesso lock) insieme alla wait, altrimenti la condizione potrebbe verificarsi tra release e wait, e il thread non verrebbe mai svegliato.

Wait va sempre usata in un ciclo: tra la generazione del segnale e la riacquisizione del lock da parte di wait la condizione potrebbe essere tornata falsa. Inoltre, in alcune implementazioni il thread può sbloccarsi anche se non viene mai chiamato signal – in questo caso la condizione non è mai stata vera.

Se signal non fosse protetto dal lock, il segnale potrebbe essere generato tra la condizione del while e la wait, quindi andrebbe perso.

Semantica Hoare

La semantica presentsata sopra è detta *Mesa*. Una versione alternativa e meno diffusa della signal (semantica *Hoare*) prevede il passaggio del lock e del controllo del processore al thread che riceve il segnale; quando il lock viene rilasciato è restituito al segnalante.

La semantica Hoare garantisce che i thread riescano ad acquisire il lock nell'ordine in cui hanno chiamato wait. Se questo comportamento è richiesto (per esempio per impedire starvation in una coda limitata), si può implementare in semantica Mesa creando una CV separata per ogni thread in attesa e inserendole in una coda. Con questo approccio si implementano facilmente anche altre politiche (e.g. LIFO), farlo con semantica Hoare è difficile.

Con semantica Hoare è sufficiente un `if` (avverarsi della condizione e `return` dalla wait sono atomici).