

Sincronizzazione

Algoritmo del fornaio

Rappresenta una soluzione al problema della sincronizzazione nel caso di N processi, ed opera secondo il seguente schema:

- prima di accedere alla propria sezione critica, un processo ottiene un numero. Fra tutti i processi che intendono accedere alla sezione critica, entra quello che detiene il numero più piccolo (come dal fornaio, appunto);
- se due processi ottengono lo stesso numero (dal fornaio non può accadere, perché vincoli fisici garantiscono la mutua esclusione; nel caso dei processi, invece, dovendo ancora risolvere il problema della mutua esclusione, non è ancora possibile escludere questa possibilità), allora accede alla sezione critica il processo con il PID minore (e questo può essere identificato in maniera univoca, essendo univoco il PID);
- lo schema di numerazione genera una sequenza di numeri non decrescente.

L'idea che sta alla base di questo algoritmo è che il problema della mutua esclusione può essere considerata come un problema di ordinamento: in pratica stabilisca l'ordine con cui i processi accedono alla sezione critica. Questo viene ottenuto distribuendo ai processi una serie di “gettoni” (*token*) numerati in ordine non decrescente ogni volta che uno di essi richiede l'accesso alla sezione critica. Tuttavia, poiché la distribuzione dei gettoni non è effettuata in mutua esclusione (infatti è proprio il problema che si sta cercando di risolvere!), non è possibile escludere che due processi ottengano un gettone con lo stesso numero (in pratica si otterrebbe un ordinamento parziale); per ottenere un ordinamento totale si usa l'identificativo del processo (questo è univoco per definizione) come criterio di secondo livello.

In dettaglio, l'algoritmo è così strutturato:

- data una coppia (*gettone*, *pid*), si definisce un ordinamento lessicografico (quello utilizzato anche per ordinare stringhe di caratteri alfanumerico) per cui si $(a,b) < (c,d)$ se $a < c$ oppure $a = c \ \&\& \ b < d$
- sono definiti due vettori di dimensione pari al numero N numero di processi
boolean *sceglie*[N] (gli elementi sono inizializzati con il valore *false*)
int *numero*[N] (inizialmente tutti gli elementi hanno il valore 0)
- l'accesso alla sezione critica per il processo i è controllato dal seguente codice

```
while (true) {
    sceglie[i] = true;
    numero[i] = max{numero[0], numero[1], ..., numero[n-1]} + 1;
    sceglie[i] = false;
    for ( j = 0; j < N; j++) {
        while ( sceglie[j] );
        while ((0 != numero[ j ]) && ((numero[ j ], j) < (numero[ i ], i)));
    }
    ...sezione critica...
    numero[ i ] = 0;
    ...sezione non critica...
}
```

Dimostrazione

Nel seguito si dimostrerà che l'algoritmo del fornaio soddisfa le 3 condizioni richieste per la soluzione al problema della sezione critica

Mutua esclusione, per assurdo

Si supponga che due processi P_h e P_k si trovino contemporaneamente nella loro sezione critica. Senza perdita di generalità, sia P_k il processo che, fra i due, entra per ultimo nella sua sezione critica (in un sistema a singolo processore, infatti, questo non può avvenire nello stesso istante per i due processi).

Quindi, se $t_{sc}^{(h)}$ e $t_{sc}^{(k)}$ rappresentano rispettivamente gli istanti in cui P_h e P_k entrano nelle loro sezioni critiche, si verifica che $t_{sc}^{(h)} < t_{sc}^{(k)}$.

Sia infine $[t_{\alpha}^{(h)}, t_{\beta}^{(h)}]$ il periodo di tempo in cui P_h seleziona il numero, ovvero $t_{\alpha}^{(h)}$ è l'istante in cui P_h inizia tale operazione, e $t_{\beta}^{(h)}$ l'istante in cui P_h termina la medesima operazione.

Ci si ponga adesso la domanda: quand'è che P_k ottiene il suo numero?

Non si può avere $t_{\alpha}^{(k)} > t_{\beta}^{(h)}$, altrimenti si avrebbe $\text{numero}[k] > \text{numero}[h]$, e P_k non potrebbe essere in sezione critica, violando la nostra ipotesi. Ma non si può nemmeno avere $t_{\beta}^{(k)} < t_{\alpha}^{(h)}$, perché P_h non potrebbe essere in sezione critica. Segue che la selezione del numero da parte di P_h e di P_k deve essere (almeno parzialmente) sovrapposta.

Quando P_h raggiunge il ciclo per confrontare il numero con quello degli altri processi, si troverà a valutare l'elemento $\text{numero}[k]$ in cui P_k ha memorizzato il suo numero (perché P_h attende che $\text{sceglie}[k]$ divenga falso); analogamente quando P_k esamina $\text{numero}[h]$. Poiché $\text{numero}[h] == \text{numero}[k]$ (altrimenti si avrebbe uno solo dei due processi in sezione critica), si deve avere che $k < h$, essendo P_k in sezione critica, e $h < k$ essendo P_h in sezione critica. Ma le due condizioni non possono essere vere contemporaneamente, per cui l'ipotesi di partenza non è ammissibile.

Avanzamento

Se uno o più processi stanno cercando di entrare in sezione critica, quello con la coppia $(\text{numero}[k], k)$ più bassa prima o poi entra (quindi almeno un processo avanza).

Attesa limitata

Quando un processo P_k raggiunge il ciclo per l'accesso alla sezione critica, il processo ha già ottenuto il numero. Gli altri processi possono eseguire la loro sezione critica al più una volta, perché quando torneranno a prendere il numero ne otterranno uno maggiore di quello ottenuto da P_k .