

L'istruzione di scambio

Oltre all'istruzione **TSL** ci sono altre istruzioni macchina che eseguono più azioni in modo atomico e possono essere usate nella soluzione del problema della mutua esclusione.

Per esempio, in alcuni sistemi è disponibile un'istruzione macchina che scambia il contenuto di due parole di memoria.



L'istruzione di scambio può essere definita formalmente con la procedura **scambia**, eseguita in modo atomico.

```
/*-----  
 * L'istruzione scambia  
 *-----*/  
void scambia(boolean x, boolean y)  
{  
    boolean temp;  
    temp = x;  
    x = y;  
    y = temp;  
};
```

La soluzione mostrata in *MutuaEsclusioneScambio* qui di seguito realizza la mutua esclusione tra processi concorrenti utilizzando la procedura *scambia*.

I processi concorrenti condividono la variabile logica *occupata*, che tiene conto dello stato della sezione critica, e usano una propria variabile privata di nome *richiesta* per indicare la volontà di accedere alla sezione critica. Si osservi con attenzione che *richiesta* in P_0 e *richiesta* in P_1 sono variabili private differenti.

```
/*-----  
 * La classe MutuaEsclusioneScambio  
 *-----*/  
class MutuaEsclusioneScambio  
{  
    static boolean occupata;  
    public static void main(String args[])  
    {  
        occupata = false;  
        /*-----  
         *  $P_0$  e  $P_1$  sono eseguiti in parallelo  
         *-----*/  
        P0();  
        P1();  
    }  
}
```

occupata: stato della sezione critica

```

/*-----*/
* P0 esegue                                P1 esegue
*-----*/
void P0()                                    void P1()
{
    boolean richiesta;                       {
    . . .                                     boolean richiesta;
    richiesta = true;                         . . .
    while (richiesta)                         richiesta = true;
    {                                          while (richiesta)
        scambia(occupata,richiesta)         {
    }                                              scambia(occupata,richiesta)
    Sezione Critica                          }
    occupata = false;                        Sezione Critica
    . . .                                     occupata = false;
}                                             . . .
}                                             }

```

Ciclo di attesa attiva

Se un processo vuole accedere alla sezione critica mette a *true* la variabile locale *richiesta*, quindi esegue l'istruzione *while* che controlla l'accesso alla sezione critica. Se la sezione critica è libera, cioè la variabile *occupata* ha valore *false*, il processo richiedente ne ottiene l'accesso dopo l'esecuzione della procedura *scambia* che mette a *true* il valore di *occupata* e a *false* il valore di *richiesta*.

Se invece la sezione critica è occupata, cioè la variabile *occupata* ha valore *true*, lo scambio di valore tra le variabili *richiesta* e *occupata* innesca un ciclo di attesa attiva che termina quando la variabile *occupata* assume il valore *false*. La variabile globale *occupata* assume il valore *false* inizialmente e quando un processo esce dalla propria sezione critica.

La soluzione è applicabile anche al caso di N processi concorrenti e, come l'istruzione **TSL**, è utilizzabile in ambiente multiprocessore. Purtroppo, anche con l'uso dell'istruzione di scambio la mutua esclusione è ottenuta fermando in un ciclo di attesa attiva il processo che aspetta l'accesso alla sezione critica.