TANKERINO.com

Lezione 194

https://www.tankerino.com/it/corsi/10/tepi-4/lezioni/194/esempio-di-errore-corsa-critica

Esempio di Errore con la Corsa Critica in C#

In questo esempio, useremo una variabile condivisa chiamata Contatore, che rappresenta il numero di operazioni completate da più thread in un programma. Supponiamo di avere due thread che incrementano il contatore contemporaneamente, senza alcun controllo di sincronizzazione. Vedremo come, a causa della corsa critica, il valore finale del contatore potrebbe non essere corretto.

Codice senza Sincronizzazione

Di seguito è riportato un esempio in C# di come due thread possano accedere a una variabile condivisa senza sincronizzazione, portando a risultati imprevisti. In questo caso, il contatore dovrebbe teoricamente arrivare a un valore di 2000, ma a causa dell'accesso simultaneo non controllato, il valore finale potrebbe essere inferiore.

```
using System;
using System.Threading;

class Program
{
    static int Contatore = 0;

    static void Main()
    {
        Thread thread1 = new Thread(IncrementaContatore);
        Thread thread2 = new Thread(IncrementaContatore);

        // Avvia entrambi i thread
        thread1.Start();
        thread2.Start();

        // Attende il termine dei thread
        thread1.Join();
        thread2.Join();
```

```
        Console.WriteLine("Valore finale del Contatore: " + Contatore);
    }

    static void IncrementaContatore()
    {
        for (int i = 0; i < 1000; i++)
        {
            Contatore++;
        }
    }
}
```

Spiegazione del Codice

Nell'esempio sopra, abbiamo una variabile statica chiamata Contatore, inizializzata a 0. Creiamo due thread, thread1 e thread2, entrambi eseguono il metodo `IncrementaContatore` in cui la variabile Contatore viene incrementata di 1 per 1000 volte.

Teoricamente, dopo che entrambi i thread hanno completato l'esecuzione, ci aspetteremmo che il valore finale di Contatore sia di 2000 (1000 incrementi per ciascun thread). Tuttavia, poiché entrambi i thread accedono alla variabile Contatore senza sincronizzazione, si verifica una corsa critica che porta a un risultato finale errato.

Come si Verifica l'Errore

Il problema principale è che i due thread stanno leggendo e incrementando il valore di Contatore contemporaneamente. Poiché il valore letto da un thread potrebbe essere lo stesso letto dall'altro, il conteggio effettivo potrebbe non riflettere correttamente il numero totale di incrementi. Questo accade perché l'operazione `Contatore++` non è atomica: consiste in tre operazioni (lettura, incremento e scrittura) che possono essere interrotte da altri thread.

L'operazione di incremento non è atomica, quindi quando due thread la eseguono contemporaneamente su una variabile condivisa, possono sovrascrivere i risultati parziali dell'altro thread, portando a errori di conteggio.

Risultato Finale

Quando eseguiamo questo codice, il valore finale di Contatore sarà quasi sempre inferiore a 2000, anche se i thread hanno eseguito l'incremento 1000 volte ciascuno. Il valore esatto varierà ogni volta

che eseguiamo il programma, a seconda di come i thread competono per l'accesso alla variabile.

Questo tipo di errore è un esempio classico di corsa critica, dove la mancanza di sincronizzazione porta a risultati imprevedibili. Per risolvere questo problema, dovremmo usare una tecnica di sincronizzazione come `lock`, che garantisce che solo un thread alla volta possa accedere alla sezione di codice che incrementa il contatore.
