

Bloccato, Libero, Sospeso



1 Thread "bloccato"

Bloccato = fermo in attesa, non può fare altro.

Esempio:

```
Thread.Sleep(3000);
```

Durante quei 3 secondi, **il thread è impegnato a dormire**.

Non può aggiornare la UI, né rispondere a input.

È come un cuoco che aspetta che l'acqua bolla guardando la pentola, senza fare altro.

→ **Bloccato** = tempo perso.



2 Thread "libero"

Libero = il thread può fare altro mentre aspetta.

Esempio:

```
await Task.Delay(3000);
```

Durante il `await`, il thread **viene restituito al sistema**.

Non rimane "fermo": può servire altri task o aggiornare la UI.

Quando il `Task` finisce, il runtime *riprende da dove aveva lasciato*.

→ **Libero** = efficiente.



3 Metodo "sospeso"

Sospeso = il metodo si ferma a metà, ma non il thread.

Esempio:

```
async Task ScaricaDati()
{
    Console.WriteLine("Inizio download...");
}
```

```
await Task.Delay(3000); // il metodo si sospende qui
Console.WriteLine("Download finito!");
}
```

Timeline:

t=0s → stampa "Inizio download..."
(il metodo si sospende)
t=0-3s → il thread è libero
t=3s → il Task termina → il metodo riprende da qui
stampa "Download finito!"

Il metodo **non continua subito**: si mette in pausa, ma il thread **torna libero**.

Quando il **Task** segnalato da **await** è completato, il runtime "riattacca" il metodo nel punto dove si era fermato.

→ **Sospeso ≠ bloccato**.

È fermo **solo lui**, non il thread su cui girava.



In sintesi

Termine	Chi si ferma?	Cosa succede
Bloccato	Il thread	Attesa attiva → inefficiente
Sospeso	Il metodo	Attesa passiva → thread libero
Libero	Il thread	Può servire altri task o aggiornare UI