

## Stati di un Thread e Diagramma degli Stati

Quando creiamo e gestiamo un thread, dobbiamo immaginare che non esegua il proprio lavoro in modo continuo dall'inizio alla fine. In realtà, ogni thread passa attraverso diversi stati nel corso della sua esecuzione. Questi stati riflettono il modo in cui il sistema operativo gestisce il thread in un dato momento.

Per esempio, un thread non può essere attivo in ogni istante, perché il sistema potrebbe avere molti altri thread da gestire. Quindi, un thread potrebbe essere in attesa di risorse, come una porzione di CPU o la disponibilità di un file. Il passaggio tra i vari stati è fondamentale per capire come funziona il multitasking.

Un thread passa attraverso vari stati che indicano il suo progresso dall'inizio alla fine della sua vita.

Ad esempio, se un thread sta aspettando di leggere dati da un file, sarà in uno stato di attesa fino a quando il file non è pronto. Poi, una volta che il file è disponibile, il thread torna in esecuzione. Questo continuo cambiamento di stato è essenziale per consentire al sistema di gestire molti thread contemporaneamente.

Immagina un thread che gestisce la stampa di un documento. Se la stampante non è pronta, il thread sarà in attesa, ma non appena la stampante sarà disponibile, il thread tornerà attivo e inizierà a stampare il documento.

### Gli stati principali di un thread

I thread attraversano diversi stati durante la loro vita. Vediamo ora i cinque stati principali che un thread può assumere.

1. New (Nuovo): Quando un thread viene creato, si trova nello stato new. Questo è il punto di partenza del thread, quando è stato definito ma non è ancora stato avviato. In questo stato, il thread esiste solo come entità nel programma, ma non ha ancora fatto nulla.

Un esempio di thread nello stato new è quando definiamo un nuovo thread in Java:

```
Thread thread = new Thread(new MyRunnable());
```

Il thread esiste, ma non è ancora stato avviato.

2. Runnable (Pronto per l'esecuzione): Una volta che il thread è stato avviato chiamando il metodo start(), entra nello stato runnable. Questo significa che il thread è pronto per essere eseguito, ma potrebbe non essere immediatamente attivo, poiché il sistema operativo deve gestire molte altre attività contemporaneamente.

Lo stato runnable indica che il thread è pronto per essere eseguito, ma potrebbe aspettare che il sistema gli assegni risorse.

3. Running (In esecuzione): Quando al thread viene effettivamente assegnato tempo di CPU, passa nello stato running. Questo è lo stato in cui il thread sta effettivamente eseguendo il proprio codice. Tuttavia, un thread può entrare e uscire dallo stato running molte volte durante la sua esecuzione, a seconda della gestione del tempo del sistema operativo.

4. Blocked/Waiting (Bloccato o in attesa): Un thread entra nello stato blocked o waiting quando sta aspettando che una risorsa diventi disponibile. Questo potrebbe essere il caso quando il thread sta attendendo l'accesso a un file, l'arrivo di un input o la disponibilità di altre risorse.

Se un thread sta aspettando che un altro thread completi un'operazione prima di proseguire, si trova nello stato di attesa. Un esempio potrebbe essere un thread che attende di leggere da un file che non è ancora disponibile.

5. Terminated (Terminato): Quando un thread ha completato il proprio compito, entra nello stato terminated. Questo significa che il thread non verrà più eseguito e ha terminato il proprio ciclo di vita.

Quando un thread entra nello stato terminated, ha completato tutte le sue operazioni e non verrà più eseguito.

## Diagramma degli stati di un thread

Per visualizzare meglio il ciclo di vita di un thread, possiamo utilizzare un diagramma degli stati. Questo diagramma mostra come un thread passa da uno stato all'altro nel corso della sua esecuzione.

Immagina una serie di cerchi, ciascuno dei quali rappresenta uno stato del thread. Le frecce tra i cerchi indicano come un thread può passare da uno stato all'altro. Ad esempio, quando un thread viene avviato, si sposta dallo stato new allo stato runnable. Da lì, quando ottiene tempo di CPU, passa allo stato running. Se deve aspettare una risorsa, può entrare nello stato blocked, e così via.

Il diagramma degli stati rappresenta il ciclo di vita di un thread, mostrando come si sposta tra i vari stati durante la sua esecuzione.

Questo diagramma è utile perché ci aiuta a comprendere come il sistema gestisce i thread e a vedere chiaramente quali sono le possibili transizioni tra uno stato e l'altro.

Un esempio pratico di come un thread può muoversi nel diagramma degli stati è quando un thread, creato ma non ancora avviato, è nello stato new. Una volta avviato, passa allo stato runnable e poi, quando il sistema gli assegna risorse, passa allo stato running. Se deve aspettare che una risorsa diventi disponibile, entrerà nello stato waiting.

## Transizioni tra stati

Le transizioni tra gli stati di un thread avvengono in base a eventi specifici. Per esempio, quando chiamiamo il metodo `start()`, il thread passa dallo stato new allo stato runnable. Allo stesso modo, quando il sistema operativo assegna CPU a un thread, questo passa dallo stato runnable allo stato running.

Le transizioni possono anche avvenire in altre direzioni. Se un thread è in esecuzione ma ha bisogno di aspettare una risorsa, come un file, entrerà nello stato waiting. Una volta che la risorsa è disponibile, tornerà allo stato runnable in attesa di essere eseguito di nuovo.

Le transizioni tra gli stati di un thread dipendono da eventi come l'assegnazione di risorse o l'attesa di input.

Questo flusso continuo tra gli stati garantisce che il sistema possa gestire molti thread contemporaneamente senza spreco di risorse.

Un thread può passare dallo stato running allo stato blocked se sta aspettando una connessione di rete. Una volta stabilita la connessione, il thread tornerà allo stato runnable in attesa di essere eseguito di nuovo.

(CC BY-NC-SA 3.0) lezione - by tankerino.com

<https://www.tankerino.com>

---

Questa lezione e' stata realizzata grazie al contributo di:



Risorse per la scuola

<https://www.baobab.school>



Siti web a Varese

<https://www.francescobelloni.it>