



Compilatore e linker nel linguaggio C



Dietro le quinte

- Il linguaggio C è un linguaggio di programmazione di basso livello utilizzato per lo sviluppo di sistemi operativi, software applicativo e altro.
- Comprendere i concetti di compilatore e linker è fondamentale per lavorare efficacemente con il linguaggio C.

Compilatore

- **Analisi Lessicale:** Il compilatore legge il codice sorgente e lo divide in token.
- **Analisi Sintattica:** Verifica la struttura sintattica del codice per assicurarsi che segua le regole del linguaggio C.
- **Analisi Semantica:** Controlla che il codice abbia un significato logico e coerente.
- **Ottimizzazione del Codice:** Migliora il codice per renderlo più efficiente.
- **Generazione del Codice:** Traduce il codice sorgente in codice macchina, creando file oggetto (.o o .obj).

Linker

- **Risoluzione dei Riferimenti:** Il linker combina i vari file oggetto generati dal compilatore e risolve i riferimenti tra moduli e librerie.
- **Creazione del File Eseguibile:** Genera un unico file eseguibile (.exe) che può essere eseguito dal sistema operativo.
- **Linking Statico vs Dinamico:** Il linking statico incorpora tutte le librerie necessarie nel file eseguibile, mentre il linking dinamico carica le librerie necessarie durante l'esecuzione del programma.

Flusso complessivo

- 1. Scrittura del Codice Sorgente:** Il programmatore scrive il codice in linguaggio C.
- 2. Compilazione:** Il compilatore traduce il codice sorgente in file oggetto.
- 3. Linking:** Il linker combina i file oggetto in un file eseguibile.
- 4. Esecuzione:** Il file eseguibile viene eseguito dal sistema operativo.