

編譯器

姓名：黃炯瑞

目錄

- | | |
|----------------|------------------|
| ❖ 簡介-----3 | ❖ 語法分析-----10 |
| ❖ 簡單流程-----4 | ❖ 語法剖析器分類—11 |
| ❖ 具體動作-----5 | ❖ 語意分析-----12 |
| ❖ 分類 -----6.7 | ❖ 優化器-----13 |
| ❖ 詞法分析與標記--8.9 | ❖ 參考資料-----14.15 |

簡介

❖ **編譯器**（compiler）是一種電腦程式，它會將某種程式語言寫成的原始碼（原始語言）轉換成另一種程式語言（目標語言）。

❖ 它主要的目的是將便於人編寫、閱讀、維護的進階電腦語言所寫作的原始碼程式，翻譯為電腦能解讀、執行的低階機器語言的程式，也就是執行檔。編譯器將原始程式（source program）作為輸入，翻譯產生使用目標語言（target language）的等價程式。原始碼一般為高階語言（High-level language），如Pascal、C、C++、C#、Java等，而目標語言則是組合語言或目標機器的目的碼（Object code），有時也稱作機器碼（Machine code）。

簡單流程

- ❖ 一個現代編譯器的主要工作流程如下：
- ❖ 將原始碼丟給預處理器做處理，再讓編譯器去做出組譯程式的動作，將其翻譯成目的碼。然後丟給連結器將目的碼相連成執行檔，最後打包好的檔案就可以給電腦去判讀執行了。

具體動作

- ❖ 詞法分析 (Lexical analysis) : 把字元合併成為詞語
- ❖ 語法分析 (Syntactic analysis) : 把詞語組合成一句有意思的句子
- ❖ 語意分析 (Semantic analysis) : 把句子組成有上文下理的段落，成為有意思的故事。
- ❖ 虛擬機 (Virtual Machine) : 用來運行編譯好的程式
- ❖ 生成代碼 (Code Generation) : 把你閱讀完的故事寫出來給虛擬機看
- ❖ 優化器 (Optimizer) : 可以把故事說得簡單一點

分類(前端)

❖ 前端主要負責解析輸入的源程式，由詞法分析器和語法分析器協同工作。詞法分析器負責把源程式中的“單詞”找出來，語法分析器把這些分散的單詞按預先定義好的語法組裝成有意義的表達式、語句、函數等。前端還負責語義的檢查，例如檢測參與運算的變數是否是同一類型的，簡單的錯誤處理。最終的結果常常是一個抽象的語法樹，這樣後端可以在此基礎上進一步優化處理。

分類(後端)

❖ 編譯器後端主要負責分析、優化中間代碼以及生成機器代碼。一般來說所有的編譯器分析、優化、變型都可以分成兩大類：函數內進行和函數間進行。很明顯，函數間的分析優化更準確，但需要更長的時間來完成。

詞法分析

❖ 詞法分析（英語：lexical analysis）是將字符序列轉換為標記（token）序列的過程。進行詞法分析的程序或者函數叫作詞法分析器（lexical analyzer，簡稱lexer），也叫掃描器（scanner）。詞法分析器一般以函數的形式存在，供語法分析器調用。

標記

❖ 這裡的標記是一個字串，是構成原始碼的最小單位。從輸入字符流中生成標記的過程叫作標記化

（tokenization），在這個過程中，詞法分析器還會對標記進行分類。

❖ 詞法分析器通常不會關心標記之間的關係（屬於語法分析的範疇），舉例來說：詞法分析器能夠將括號識別為標記，但並不保證括號是否匹配。

❖ `sum = 3+2;`

❖ 將其標記化後可以得到下表內容：

❖ `sum`：標示符

❖ `=`：賦值操作符

❖ `3`：數字

❖ `+`：加法操作符

❖ `2`：數字

❖ `;`：語句結束

語法分析

- ❖ 根據某種給定的形式文法對由單詞序列構成的輸入文字進行分析並確定其語法結構的一種過程。
- ❖ **語法剖析器**通常是作為編譯器或直譯器的組件出現的，它的作用是進行語法檢查、並構建由輸入的單詞組成的資料結構。語法剖析器通常使用一個獨立的詞法剖析器從輸入字元流中分離出一個個的「單詞」，並將單詞流作為其輸入。實際開發中，語法剖析器可以手工編寫，也可以使用工具（半）自動生成。

語法剖析器分類

有頂向下

❖ 根據形式語法規則，在語法分析樹的自頂向下展開中搜尋輸入符號串可能的最左推導。單詞按從左到右的順序依次使用。

由下而上

❖ 語法剖析器從現有的輸入符號串開始，嘗試將其根據給定的形式語法規則進行覆寫，最終覆寫為語法的起始符號。

語意分析

❖ **語意分析**技術是指將一長串的文字或內容，從其中分析出該個段落的摘要以及大意，甚至更進一步，將整篇文章的文意整理出來。此項技術可以應用在解讀影片、音訊等檔案，使得搜尋引擎能夠搜尋到文字以外的物件，方便使用者省去大量時間觀看影片、聆聽音訊，同時也可以幫助使用者提前了解影片與音訊的內容。

優化器

- ❖ 把程式碼做一個減少的動作，每一行程式即便跟最終的執行結果並不會帶來太大的影響，但是電腦還是會需要吃一些電腦的資源去讀取這一段毫無相關的程式碼。
- ❖ 達到優化的最簡單的方式有刪掉沒有用過的變數。又或者將比較長、雜的程式，替換成簡短或耗用的資源較少的程式碼。


參考資料

- ❖ 編譯器-維基百科 <https://zh.wikipedia.org/zh-tw/%E7%B7%A8%E8%AD%AF%E5%99%A8>
- ❖ 編譯器-MBA智庫 <https://wiki.mbalib.com/zh-tw/%E7%BC%96%E8%AF%91%E5%99%A8>
- ❖ 深入淺出教你寫編譯器 <http://www.inspiregate.com/programming/other/471-compiler-1.html>
- ❖ 詞法分析 (Lexical analysis) <https://zh.wikipedia.org/zh-tw/%E8%AF%8D%E6%B3%95%E5%88%86%E6%9E%90>
- ❖ 語法分析 (Syntactic analysis) <https://zh.wikipedia.org/zh-tw/%E8%AF%AD%E6%B3%95%E5%88%86%E6%9E%90>

參考資料

❖ 語意分析 (Semantic analysis)

<https://zh.wikipedia.org/zh-tw/%E8%AA%9E%E6%84%8F%E5%88%86%E6%9E%90>



結 束