

序

系統程式在電子資訊產業界是如此的重要，但在學術界卻不太受到重視，強勁的產業需求與疲弱的人才供給之間，形成了一大段的市場落差，作者希望能藉由本書，縮小這段落差。

系統程式課程的教科書，大致上分為兩類，第一類是以系統軟體理論 (System Software) 為主的書籍，而第二類是以系統程式 (System Programming) 實務為主的書籍。

以系統軟體理論為主的書籍當中，最著名的書是 Beck 的 *System Software: An Introduction to Systems Programming*。這本書以 SIC/XE 這個假想的機器出發，很清楚的說明了系統軟體的概念，並且理論陳述上掌握得相當好，既簡單又清楚。然而、該書已有 20 年多年的歷史，且以 Pascal 與 SIC/XE 的組合語言為範例，忽略了現今以 C 語言為核心的產業現況，導致該書完全與實務脫節，讓學生無法理解真實世界的狀況。

在以實務為主的系統程式書籍中，通常分為很多子類。像是『Linux 系統程式設計』或『Windows 系統程式設計』之類的書，會將焦點鎖定在單一作業系統上。這種書籍在實務上很有用，但是卻很容易因為作業系統的變化而過時。況且，這類書籍的焦點是作業系統，所以通常不會納入組譯器、連結器、載入器、巨集處理器、編譯器等主題，因此不適合作為系統程式課程的教科書。

找不到適當的教科書，是系統程式課程傳授者的困擾之一，另一個困擾是到底要教授甚麼主題，也還處於眾說紛紜的狀況。舉例而言，有些學校將『系統程式與組合語言』合併成一門課程，有些則分開。

組合語言雖是系統程式的主題之一，但卻只是其中的一小部分，無法涵蓋系統程式的一些重要部分，像是編譯器、組譯器、連結器、作業系統等等。另外，C 語言在系統程式中的重要性，並不亞於組合語言，甚至有過之而無不及，這些都不是組合語言課程所能涵蓋的。

因此，本書捨棄此種作法，直接將『系統軟體』與『系統程式』兩者都納入『廣義的系統程式』範圍當中，以便較完整的涵蓋『系統軟體』與『系統程式設計』這兩個密切相關的領域，以避免遺漏了重要的主題。

我們期望藉由同時納入『系統軟體』與『系統程式設計』等兩個密切相關的主題，達到相輔相成的目的。因為我們相信，學習系統程式，除了可以幫助學習者理解電腦運作的原理之外，還能在實務上培養出優秀的系統程式設計師。我們認為，這兩個定義都是系統程式課程的重點，不可偏廢。

在本書中，我們假定讀者已經學過了至少一門的程式設計課程，如果讀者學習過 C 語言，那對於閱讀本書將會有很大的幫助，我們將採用組合語言與 C 語言等兩個主角，闡述系統軟體的理論與系統程式的實務，讓讀者得以透過這兩個語言，快速的掌握系統程式的精隨。

然而，本書的篇幅絕對無法涵蓋系統軟體與系統程式的所有面向，因此，我們將捨棄某些可能被納入系統程式的主題，像是資料庫系統、軟體工程等，以便使本書能聚焦在典型的系統軟體與系統程式兩個領域，進行較深入的闡述。

筆者認為，系統軟體與系統程式兩者，對資訊工程系以及電子科系的學生而言都相當重要。因此，為了較為全面且平衡的介紹『系統軟體』與『系統程式』兩個主題，我們將採取兩者並重的方式。

在系統軟體方面，我們利用了一個簡化版的處理器 CPU0，說明組譯器、連結器、載入器、虛擬機器與電腦硬體의運作原理。透過簡化的處理器，我們可以避免掉實務上的複雜性，直接深入電腦背後的原理，以避免被太複雜的機器架構與組合語言所迷惑。

在系統程式方面，我們主要採用開放原始碼的 GNU 開發工具，以實際操作的方式學習『系統程式實務』，這可以讓我們不至於脫離現實太遠，也能讓讀者順便熟悉 GNU 這個在產業上具有無比重要性的工具集。

本書將從第 1 章的『系統軟體』出發，然後在第 12 章以『系統軟體實作』結尾。中間的章節分為三個部分，第一部分是 2-6 章的『組合語言』相關主題，第二部分是 7-8 章的『高階語言』相關主題，而第三部分則是 9-11 章的『執行平台』相關主題。

在『組合語言』部分，我們從第 2 章的『電腦的硬體結構』出發，然後在第 3 章進入『組合語言』領域。接著，導入組合語言相關的系統軟體，這包含『組譯器』(第 4 章)、『連結與載入』(第 5 章) 等主題。

然後，利用巨集處理器（第 6 章）開始導入高階語言，巨集處理器是一個組合語言與高階語言都會用到的工具，可以讓程式語言更有彈性。

在『高階語言』部分，我們從第 7 章的『高階語言』出發，然後說明高階語言的重要系統軟體 - 『編譯器』(第 8 章) 是如何製作的，並且以 C 語言作為主要對象，說明其語法與編譯器的設計原理。

在『執行平台』部分，我們以第 9 章的『虛擬機器』開頭，然後在第 10 章學習『作業系統』的主題，最後以原始機器的『嵌入式系統』作為結尾，讓讀者感受平台與程式之間的關係，並且深入理解平台背後的设计原理。

最後，在第 12 章中，我們將實作出小型的『組譯器』、『編譯器』以及『虛擬機器』，以便分別代表『組合語言』、『高階語言』與『執行平台』三類系統軟體，讓讀者實際體會其設計方法。

筆者認為，系統程式並非單獨的一門課程，而是一個整合『數位邏輯』、『計算機結構』、『嵌入式系統』、『作業系統』、『編譯器』等課程的核心領域，本書希望能在化繁為簡的原則上，建立系統程式與上述課程之間的關係。

本書的網路資源

在筆者撰寫本書的過程中，翻閱了無數的書籍，並查詢了難以計數的網路文章。要將這麼龐大的領域納入到一本書當中，實在是難以達成的任務，最後，筆者捨棄的文章似乎比納入到本書中的文章更多，為了避免有遺珠之憾，筆者為本書開闢了一個專屬網站 <http://sp1.wikidot.com>，以便讓讀者能進一步翻閱這些捨棄後的文章，進一步學習系統程式的各個領域。

您可以從該網站中取得本書的補充教材、範例程式與投影片等資訊，我們希望透過該網站，能隨時讓讀者取得最新的相關訊息，並且隨時更新書籍中的錯誤，讓您的系統程式知識能隨著網站的更新而成長，也讓筆者與讀者之間有一個好的溝通橋梁。

在本書的光碟中，我們附上了本書中的範例程式，包含 C 語言與 CPU0 的組合語言等範例，也包含了第 12 章中筆者所實作的系統軟體，包含編譯器 (c0c)、組譯器 (as0) 與虛擬機器 (vm0) 的原始程式碼，以供讀者閱讀與測試時使用，請讀者在閱讀相關章節時，能夠充分利用這些範例程式，以便更深入的理解系統軟體的設計原理。

感謝

長久以來，筆者在教授『系統程式』課程時，總是感到無比困擾，這個困擾來自於找不到適當的教科書。當筆者採用理論性的書籍時，學生會感覺不到系統程式在實務上的重要性，但是採用實務性的『組合語言』書籍時，卻又造成領域太窄與名實不符的情況，於是我一直希望能有人寫一本理論與實務兼具的系統程式教科書，但卻又遍尋不到。

於是我寫了一份相當簡短的草稿放在自己的網站上，旗標出版社的彥發兄看到該草稿後，便主動與我聯絡，希望能將該草稿出版。這下讓我既欣慰又感到壓力，欣慰的是終於有系統程式的教科書可以用了，但這個任務卻要由我自行完成，因此而倍感壓力。

現在，這本書終於完成了，我得感謝旗標的彥發兄與昕晔兄，沒有彥發兄的提議，我可能一輩子都無法將此書完成。而沒有昕晔兄的修改校對，這本書恐怕會難以閱讀。

另外，也感謝旗標公司中無數參與這本書的同仁，沒有你們的參與，這本書將無法付梓出版。當然，對於我個人而言，更應該感謝的是我的家人，我的太太與兩個小寶貝，為了這本書而犧牲了好多個周末的時光，以至於無法好好的陪伴你們，感謝妳們的支持、關心與照顧。筆者誠心的感謝每一個曾經幫助過我們的人，希望大家都能平安、健康且快樂，祝福大家！

陳鍾誠 2010 年 5 月 於金門大學