

# 系統程式

(為何撰寫此書)

作者：陳鍾誠

旗標出版社

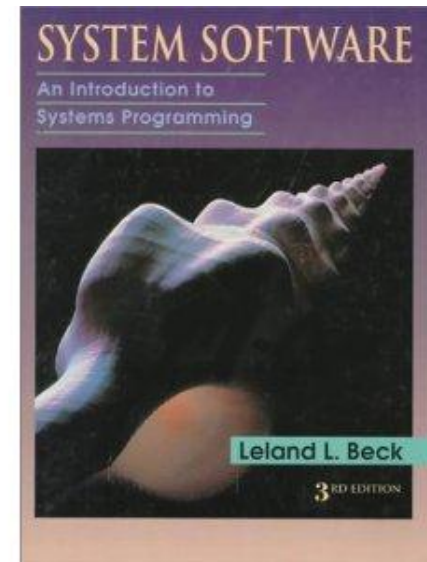
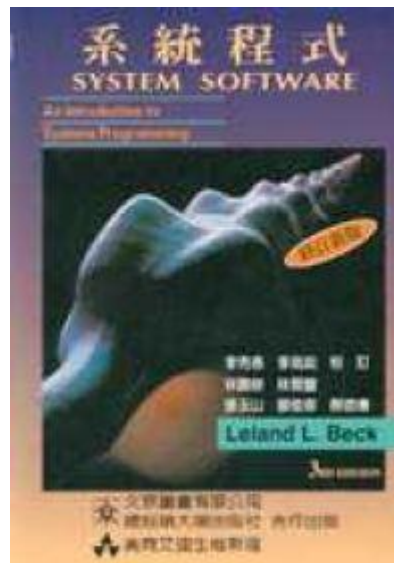


# 系統程式 – 舊約聖經

- 25 年來，大部分的「系統程式」教科書都是採用 Beck 的 “System Software: An Introduction to Systems Programming,” 一書，這本書可以說是系統程式課程的聖經。
- 該書由於以貝殼為封面，作者又是 Beck，因此被簡稱為貝殼書。
- 簡單，理論清楚

# 貝殼書

- 目前最常被採用的教科書是俗稱貝殼書的
  - System Software An Introduction to System Programming 3rd, Leland L. Beck, Addison Wesley. 台北圖書代理.
  - 系統程式 第三版, 文京圖書有限公司, 艾迪生維斯理出版社.



# 貝殼書為何成為聖經

- 優點：簡單，理論清楚
- 沒有競爭對手
  - 理論類書籍 – 大部分是考試用書
    - 考試用書：翟高壽 (新文京)、林建福 (儒林)
    - 教科書：許明壽 (松崗)，林煙桂 - 使用 Turbo C
      - 寫得不夠好，而且已過時
  - 實作類書籍 – 不適合作為教科書
    - 系統程式實作 - 林邦傑 (林煙桂應該就是林邦傑)
    - DOS 系統程式設計 -- (1980 – 1990) 台灣以施威銘先生最具代表性
    - Linux 系統程式設計 – 很多 (Jollen 陳俊宏, ....., 邱毅凌, Jserv, ...)
    - Windows 系統程式設計 – 也不少

# 貝殼書的章節架構

- 第一章 背景
- 第二章 組譯器
- 第三章 載入器和連結器
- 第四章 巨集處理器
- 第五章 編譯器
- 第六章 作業系統
- 第七章 其他系統軟體
- 第八章 軟體工程導論
- 附錄A SIC/XE 指令集與位址模式
- 附錄B ASC II 字元碼
- 附錄C SIC/XE 參考資訊

# 本書與貝殼書的章節比較

- 第一章 背景
- 第二章 組譯器
- 第三章 載入器和連結器
- 第四章 巨集處理器
- 第五章 編譯器
- 第六章 作業系統
- 第七章 其他系統軟體 (資料庫)
- 第八章 軟體工程導論
- 附錄A SIC/XE 指令集
- 附錄B ASC II 字元碼
- 附錄C SIC/XE 參考資訊
- 第 1 章 系統軟體
- 第 2 章 電腦的硬體結構
- 第 3 章 組合語言
- 第 4 章 組譯器
- 第 5 章 連結與載入
- 第 6 章 巨集處理器
- 第 7 章 高階語言
- 第 8 章 編譯器
- 第 9 章 虛擬機器
- 第 10 章 作業系統
- 第 11 章 嵌入式系統
- 第 12 章 系統軟體實作
- 附錄 A CPU0 處理器
- 附錄 B C0 語言的語法
- 附錄 C GNU 開發工具
- 附錄 D Dev C++ 開發環境
- 附錄 E Cygwin 開發環境

# 貝殼書的優缺點

- 優點
  - 簡單
    - 使用 SIC/XE 簡化處理器，避開了實際處理器的複雜性
  - 理論清楚
    - 以 SIC/XE 貫穿整本書，清楚的闡述了組譯、連結、載入、編譯等主題。
- 缺點
  - 沒有使用 C 語言 (Beck 先生熟悉的是 Pascal)
  - 沒有實作系統軟體
    - 通常撰寫 SIC 組譯器是這門課學生的作業
    - 作者曾經提供 Pascal 寫的 SIC 組譯器，但很少人用了 (因為 Pascal)
  - 沒有操作實務
    - 貝殼書只有理論。

# 貝殼書為何能風行 25 年？ (1)

- 25 年前

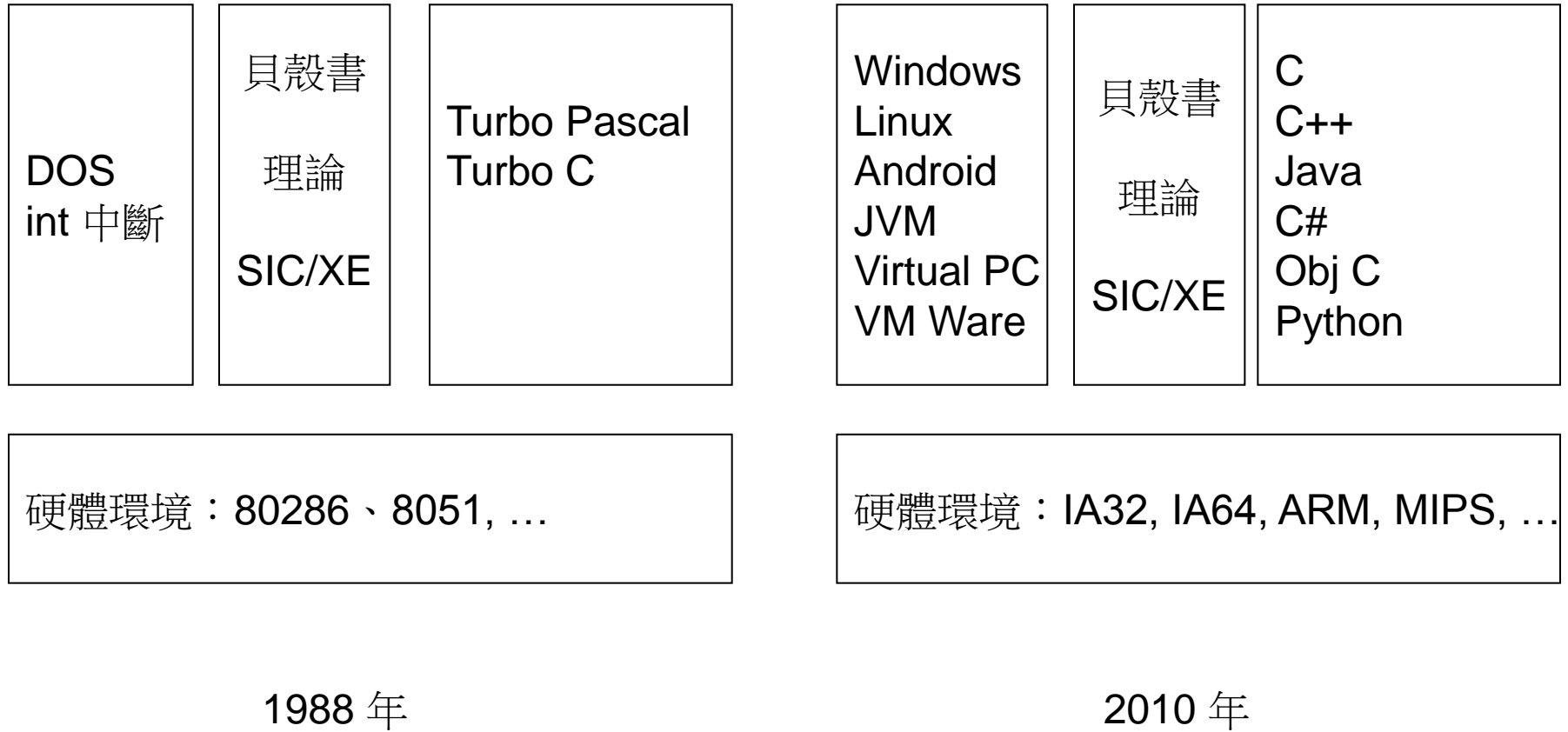
- 硬體環境：80286, 8051, ...
- 作業系統：DOS 時代
- 程式環境：使用 Turbo Pascal, Turbo C, ....
- 程式語言：Pascal, C, ...

- 現在

- 硬體環境：IA32, ARM, MIPS, ...
- 作業系統：Windows, Linux, Android, OS X, ...
- 程式環境：Visual Studio, Eclipse, Dev C++, GNU, ...
- 程式語言：C, C++, C#, Java, Obj C, Python, ....



# 貝殼書為何能風行 25 年？ (2)



# 一個迷惘了 20 年的資訊系學生

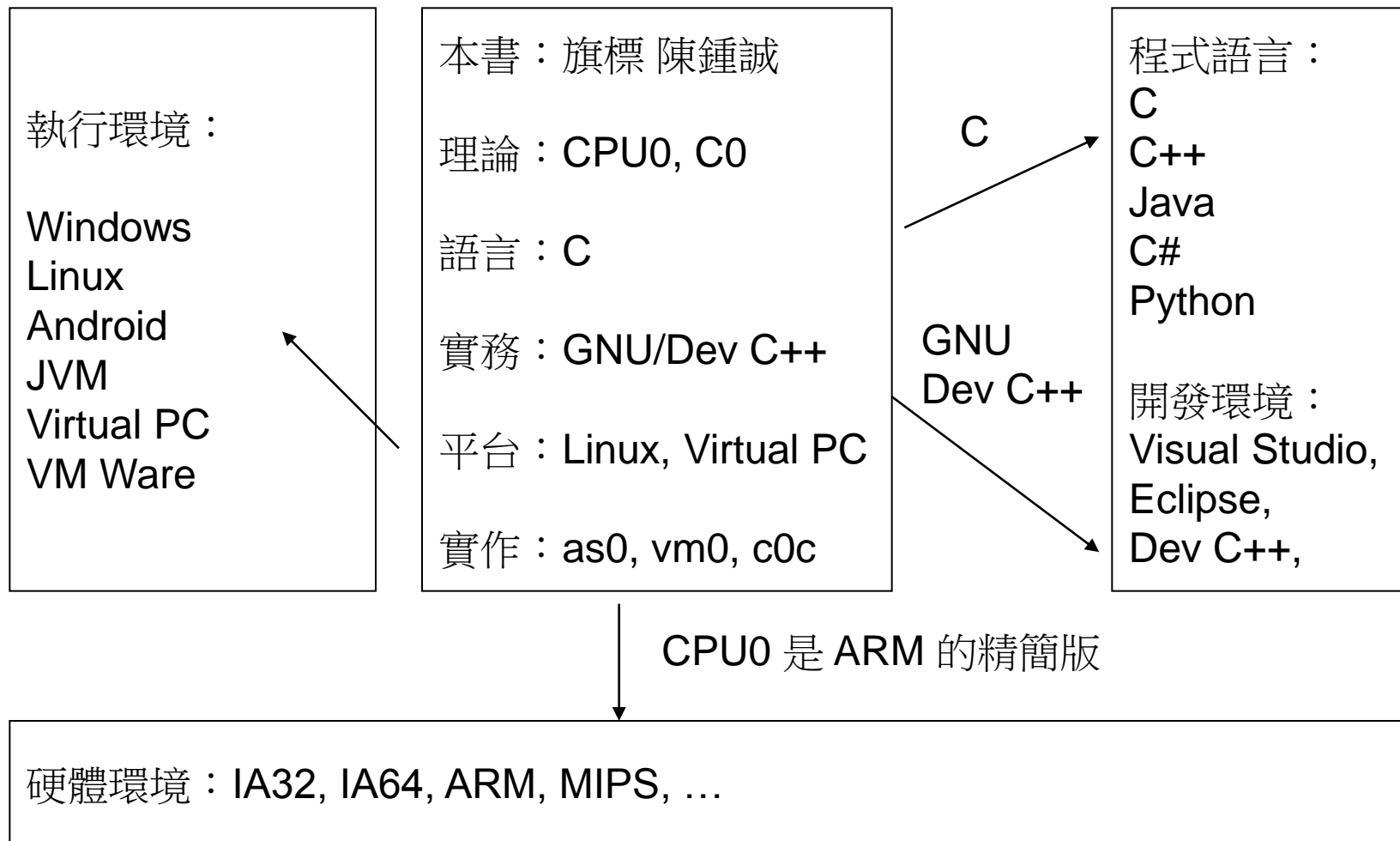
- 1989 年

- 我在交通大學唸書時，老師就是使用這本書作為教材，當時對這門課似懂非懂，雖然我很認真的在課堂上寫出一個 SIC 機器的組譯器，但是仍然認為這門課與產業實務無關，我未來應該用不到它。

- 2002 年

- 進入產業界之後，我才發現原來「系統程式」並不是一種理論，而是整個台灣電子資訊產業的核心。整整經過 20 年，我才恍然發現這門課的用途。環顧台灣的電子資訊產業，亟需要大量的系統程式人才，但是 Beck 的書籍所提供的內容，無法讓學生認識到這個情況，因為只有理論而沒有實務。

# 本書的環境基礎



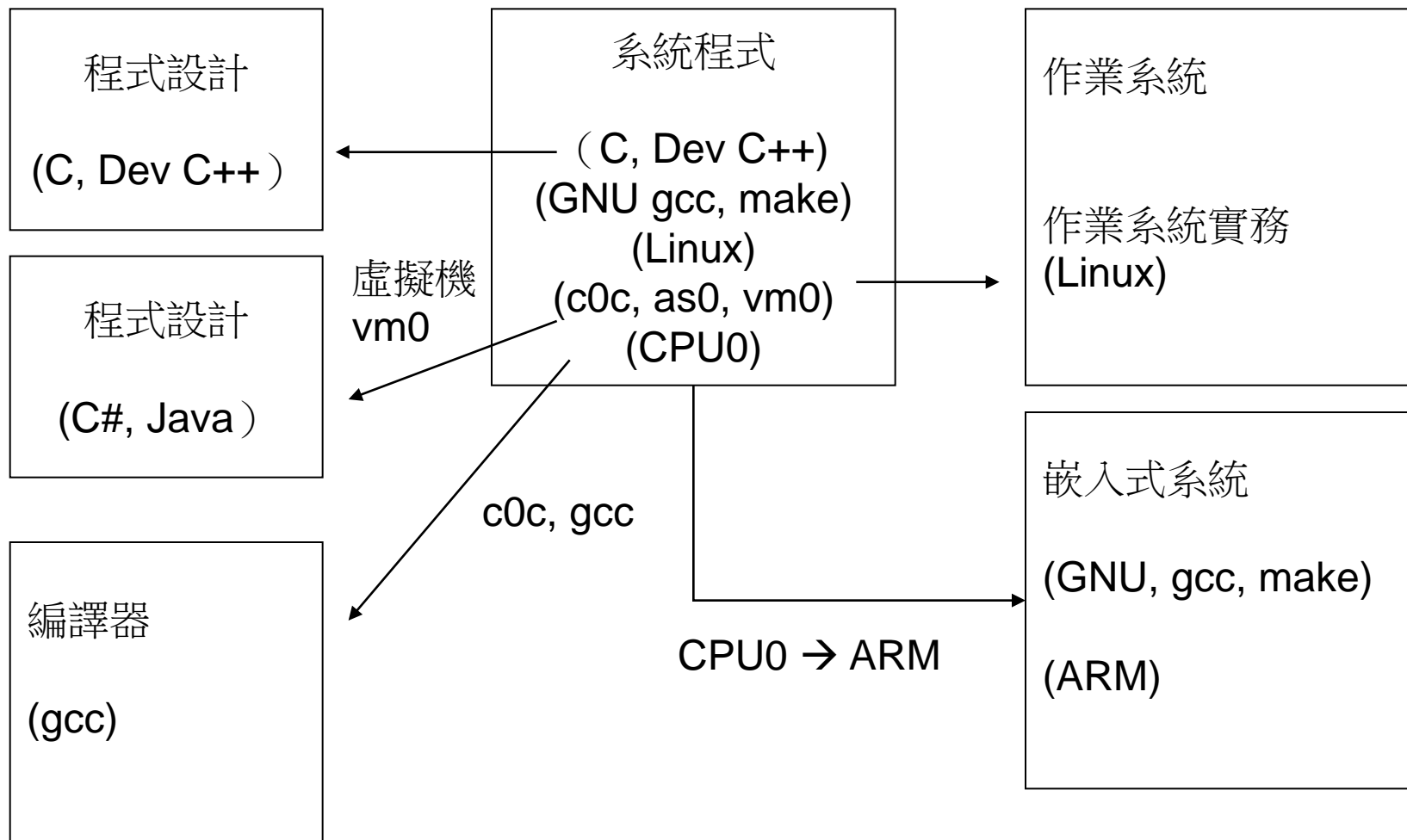
# 撰寫本書的原因

- 核心原因：
  - 「系統程式」並不只是一種理論

# 本書的特色（重點所在）

- 理論：採用簡化的處理器
  - 承襲貝殼書的方法，但將 SIC/XE 改為更接近 ARM 的 CPU0
- 語言：以 C 語言為核心
  - C 語言在「系統程式、嵌入式系統、作業系統」上極為重要
  - 清楚說明 C 語言與組合語言的連結方式
- 實作：採用 C 語言實作系統軟體
  - 實作組譯器 (as0)、虛擬機 (vm0)、編譯器 (c0c)
- 實務案例：結合今日的工具與環境
  - 以 GNU 工具為主：gcc, as, ld, make, objdump, ...
  - 使用學生最常用的工具 Dev C++
  - 結合 Linux 作業系統與 Virtual PC 虛擬機

# 本書與其他課程的關係



# 本書的網站

- 網址:

- <http://sp1.wikidot.com/>

- 內容

- 投影片下載
- 程式碼下載
- 大量的補充教材，隨時新增與修改
  - 微軟組合語言
  - 微軟開發工具：Visual Studio, ML, CL, LINK, ...
  - Linux 作業系統：行程切換、Thread、
  - 理論補充：Thread、競爭情況、死結、號誌 (以 C# 示範)
  - 產業訊息：Android, QEMU, 嵌入式裝置, ...

# 未來

- 程式、程式、程式
- 重新打造輪子
  - 已有：編譯器 (c0c)、組譯器 (as0)、虛擬機 (vm0)
  - 目前：
    - 作業系統：os0 – 以 cpu0 的組合語言寫成
  - 未來
    - 處理器：cpu0 – 以 VHDL 實作，在 Altera FPGA 板上設計運行，使用 Quartus II 模擬
- **KISS: Keep it Simple and Stupid**



# 出版計劃

- 用程式打造自己的電腦 – 從軟體到硬體
  - 編譯器 c0c (使用 c 語言)
  - 組譯器 as0 (使用 c 語言)
  - 虛擬機 vm0 (使用 c 語言)
  - 作業系統 os0 : 使用 c0 語言 + cpu0 的組合語言
  - CPU0 : (使用 VHDL 語言)
    - 開發時使用 Altera Quartus II 進行模擬
    - 完成後使用 Altera FPGA 板進行燒錄測試

# 後記 (1)

- 直到我接受彥發兄的邀請，開始撰寫此書之後，我突然明白了一件事
  - 問題：
    - 為何沒有人願意寫一本全新的系統程式教科書
  - 答案：
    - 這個主題真的是太廣大、太複雜了，要懂得東西很多，又超級難寫。

## 後記 (2)

- 修正完本書交稿給昕暉兄之後，我就住院了
  - 感謝昕暉與彥發兄特別到醫院看我
  - 看來我被系統程式難倒了，難到頭都暈了