

序

我於 1997 年任職一家硬體廠商的時候，首次提議在嵌入式系統中使用 Linux，我的建議在當時不僅令人感到驚訝，也受到某些質疑。今日，在嵌入式系統中使用 Linux 不再是可笑之事。的確，有不少大公司及政府機關在其嵌入式軟體的需求上逐漸仰賴 Linux。

Linux 在一些嵌入式應用的成功，引發廣泛的興趣和熱烈的迴響，這導致了「嵌入式 Linux」相關文章、網站、公司及文件的氾濫。除了一閃即逝的宣佈，雜誌裡的文章，數以百計的專案和產品，都在在宣告了將 Linux 使用於嵌入式系統的輕而易舉，然而正在探索有用指引的專業開發者們，卻仍在為建立嵌入式 Linux 系統的基本方法和技術尋覓解答。

目前大部分的文件都只著重在，如何使用各類套裝即用的跨平台開發工具，以及目標板二元碼 (target binary)【譯註】。有的文件則只侷限在如何在目標板 (target board) 上執行 Linux。

與現有的文件不同，本書並不會假定你手邊有哪些工具，或是限定專案的範圍。本書只會要你透過 Internet 下載必要的套件、瀏覽特定的線上文件，以及透過專案的郵遞論壇 (mailing list) 與其他開發者一起分享經驗。你還需要一部開發主機以及目標板硬體的說明文件，但我不會要求你向任何廠商購買任何產品或服務。

●.....

譯註 目標板 (target) 係嵌入式系統的代名詞，通常用於軟體開發期間，用來區分嵌入式系統與開發主機。目標板二元碼 (target binary) 係指目標板 (target board) 上以二元 (0 與 1) 形式存在記憶體中供 CPU 執行的機器碼 (machine code) 或目的碼 (object code)。

除了提供最大的自由度，以及讓你掌控自己的設計之外，這麼做最接近帶頭在嵌入式系統中使用 Linux 的先驅者們所採行的方式。這些先驅者對 Linux 所做的事，本質上包括將它拆開以符合應用的需要，以及根據其目的對它進行客製化。因此，Linux 突破 embedded 世界的方式不同於許多軟體廠商將其產品擴展至新應用領域的做法。身為嵌入式系統開發者的你可能會發現，相較於廠商所販賣的產品，Linux 比較容易應用到你的設計中。

本書的做法就是提供所有的細節，以及討論在嵌入式系統中使用 Linux 時將會遇到的各種困境，讓你得以把 Linux 應用在設計中。雖然本書的內容不可能涵蓋所有的嵌入式設計，但是書中所提供的資源卻可讓你輕易獲得必要的資訊，使你得以在嵌入式系統中使用 Linux，並對它進行客製化。

撰寫本書的目的在拉近「於設計中使用開放原碼和自由軟體之嵌入式系統開發者」跟「建立和維護這些開放原碼和自由軟體套件之開發者」的距離。儘管有許多主流嵌入式系統開發者（其中不乏高水準的程式設計者）會仰賴協力者（third-party）提供其所需要的嵌入式 Linux，無疑地他們也有機會對其所仰賴的開放原碼（open source）【譯註】和自由軟體（free software）計劃做出貢獻。最後，這種動力將可確保 Linux 持續成為嵌入式系統首選的作業系統。

本書對象

本書鎖定的第一種讀者，是那些想要在未來或目前的計劃中使用 Linux 之有經驗的嵌入式系統設計者。本書會假定這一類讀者已經熟悉用於開發嵌入式系統的技術和術語，像是交叉編譯（cross-compiling）、進行 BDM 或 JTAG 除錯，以及如何處理不完善或不完全的硬體。如果你是這一類的讀者，你可能會想要跳過前面一些章節中，與嵌入式系統開發背景知識有關的部分。然而，你還是需要閱讀一些前面的章節（尤其是第 2 章），因此這類章節包含了在嵌入式系統中使用 Linux 核心的特別提示。

本書鎖定的第二種讀者，是那些想要熟悉嵌入式 Linux 系統開發工具和技術之無經驗的嵌入式系統開發者。本書並非嵌入式系統的入門書，不過若你想探究本書所討論到的一些課題，你可以參考一些入門性質的教科書。本書附錄 B 表列了一些有用的書籍以及資訊來源。

●.....
譯註 「source」或「source code」可譯成「原始碼」或「來源碼」。若為前者，則可簡稱為「原碼」；若為後者，則可簡稱「源碼」。

如果你是個對 Linux 已經很熟悉的玩家 (power user) 或系統管理者，本書應該可以協助你進行高度客製化的 Linux 安裝。例如，若你發現到發行套件 (distribution) 安裝了過多的套件 (package)，而你想要從頭開始建立自己的客製發行套件 (custom distribution)，本書有許多部分應該可以派上用場，特別是第 6 章的內容。

最後，本書應該可以協助那些想要瞭解如何建立與操作 Linux 系統的程序設計者或 Linux 的熱衷者。儘管本書並未說明如建立一般用途發行套件 (general-purpose distribution)，不過其所提到的許多技術，在一定程度上可以媲美用來進行客製化嵌入式 Linux 安裝的一般用途發行套件。

本書範疇及背景知識

要在嵌入式系統中展現 Linux 的最佳能力，你需要以下背景知識，這些知識在許多書中都交代的很清楚：

嵌入式系統

一般而言，你必須熟悉嵌入式系統的開發、程式設計和除錯，這包括軟體、硬體兩方面。

Unix 系統管理

你必須能夠勝任各種的系統管理工作，像是硬體的組態、系統的設定、維護以及利用 shell 命令稿讓管理工作自動化。

Linux 裝置驅動程式

你必須知道如何對各種 Linux 裝置驅動程式進行開發及除錯。

Linux 核心的運作原理

你必須盡可能瞭解核心的運作原理。

GNU 軟體開發工具

你必須具備有效利用 GNU 工具的能力。這包括瞭解許多常被認為是晦澀難懂的選項和公用程式。

本書會假定你至少熟悉基本概念中所論述的每個項目。另一方面，閱讀本書你無須知道，例如 Linux 裝置驅動程式是如何建立的，或與嵌入式系統開發有關的每一件事。翻閱本書的時候，若看到符合你的嵌入式系統的 Linux 用法，你可能會覺得需要取得這個 Linux 用法的進一步資訊。除了閱讀本書時你自己參考的其他書籍，還可以看一附錄 B 所列的書單，你或許會因而找到這些背景知識的進一步資訊。

儘管本書只討論如何在嵌入式系統中使用 Linux，但是對想要在嵌入式系統中使用 BSD 的開發者來說多少會有一些幫助。不過本書所作的許多說明，都必須依據 BSD 與 Linux 間的差異重新詮釋過。

本書架構

本書由三大部分構成。第一部分由第 1 到 3 章組成。這三章的內容涵蓋建立任何嵌入式 Linux 系統必備的基本知識。儘管這三章並未提到任何實作程序，不過它們都是建立嵌入式 Linux 系統時不可獲缺的知識。

第二部分由第 4 到 9 章組成。這幾章的內容都非常重要，其中安排了用來建立任何嵌入式 Linux 系統的必要步驟。不管你的系統的目的或機能是什麼，都必須閱讀這幾章的內容。

第三部分由第 10 和 11 章組成。雖然這兩章所包含的內容都很重要，不過並非建立嵌入式 Linux 系統的要素。

第一章《入門指引》將會深入嵌入式 Linux 系統的世界。本章首先會介紹基本定義，接著會從實務的觀點來探討嵌入式 Linux 系統的議題，包括從嵌入式系統的觀點來探討開放原碼（open source）和自由軟體（free software）版權。然後會介紹本書其他部分將會用到的範例系統，以及本書將會用到實作方法。

第二章《基本概念》用來描述建立任何嵌入式 Linux 系統時所必須具備的基本觀念。

第三章《所支援的硬體》完整地介紹了 Linux 所支援的嵌入式硬體，以及告訴讀者何處可以找到實作這些硬體的驅動程式（driver）和子系統（subsystem）。本章還會探討處理器的架構、匯流排和介面、I/O、儲存裝置、一般用途的網路功能、工業等級的網路功能，以及系統監控。

第四章《開發工具》將會說明如何安裝與使用各種用來建立嵌入式 Linux 系統的開發工具。其中最值得注意的是，如何從頭開始建立與安裝 GNU toolchain 元件。此外還會分節討論 Java、Perl 和 Python，以及探討可用來與嵌入式目標板互動的各種終端機模擬程式。

第五章《核心方面的考量》將會探討在嵌入式系統中 Linux 核心的選擇、組態、交叉編譯、安裝以及使用。

第六章《根檔案系統的內容》將會說明如何使用本書稍早所提到的元件來建立根檔案系統，包括 C 程式庫的安裝，以及建立適當的 `/dev` 項目。比較值得注意的是，本章的說明涵蓋了 BusyBox、TinyLogin、Embutils 和 System V *init* 的安裝和使用。

第七章《儲存裝置的操作》將會解說如何操作以及為嵌入式 Linux 系統設置儲存裝置。本章將重點擺在固態（solid-state）儲存裝置上，像是原生的 flash 和 DiskOnChip 裝置，以及 MTD 子系統。

第八章《根檔案系統的規劃》將會說明如何為嵌入式系統的儲存裝置設定第六章所建立的根檔案系統。這包括建立 JFFS2 和 CRAMFS 檔案系統的映像，以及經由 NFTL 使用磁碟形式的檔案系統。

第九章《設置 Bootloader》將會針對每個嵌入式 Linux 平台探討各種可用的 bootloader（開機載入程式）。並且會進一步探討如何在 DiskOnChip 裝置上使用 GRUB 以及 U-Boot。此外還包含以 BOOTP/DHCP、TFTP 和 NFS 來進行網路開機。

第十章《設置網路服務》會將重點擺在各種網路服務（像是 SNMP、SSH 和 HTTP）之軟體套件的組態、安裝和使用。

第十一章《除錯工具》的內容涵蓋為嵌入式 Linux 系統開發軟體時可能遭遇到的主要除錯議題。這包含了如何在跨平台（cross-platform）開發環境中使用 *gdb*、進行追蹤、效能分析，以及記憶除錯。

附錄 A《工作單》提出了一份可搭配本書一起使用的工作單，可用以為嵌入式 Linux 系統提供完整的計劃書。

附錄 B《資源》的內容包含了建立嵌入式 Linux 系統時可供利用的資源。

附錄 C《重要的版權聲明》的內容包含了 Linus Torvalds 與其他核心開發者討論核心版權以及非 GPL 版權之二元核心模組相關議題時的重要信件。

儘管第 7 到 9 章看起來像是各自獨立的章節，不過它們的內容卻有高度的關連性。舉例來說，要瞭解第 7 章所探討之目標板儲存裝置的設定，得先瞭解第 8 章所探討之目標板檔案系統，反之亦然。此外，儲存裝置的設定還必須先瞭解第 9 章所探討之 bootloader 的設定與操作，反之亦然。因此，我建議各位先一口氣讀完第 7 到 9 章，再執行這幾章所提到的任何指令。這樣當你設定目標板的時候，才有辦法按照這幾章所描述的程序進行操作。

本書所用到的硬體

正如我們在第 3 章所見，Linux 支援了各式各樣的硬體。由於篇幅有限，本書只能選用幾種嵌入式系統作為各種程序的測試平台。表 P-1 完整地列出了本書將會用到的系統。

這些系統中有些是主流市場上的商品，像是 iPAQ 或是 Dreamcast。我刻意將它們列入，主要是讓有興趣的讀者可以輕易找到用來學習建立嵌入式 Linux 系統的材料。例如，Sega 出品的 Dreamcast 遊戲機，可以在 eBay 上以低於 50 美元的價格買到。儘管 Dreamcast 使用格式特殊的 CD-ROM 來開機，不過它卻是學習 Linux 跨平台開發的最便宜方法之一。當然，也可以使用老舊的 x86 PC 來作實驗，不過這樣比較沒意思就是了。

表 P-1：本書所用到的目標板系統

架構	系統類型	處理器的時脈速度	RAM 的大小	儲存空間的大小及類型
PPC	TQ components TQM860L	80 MHz	16 MB	8 MB flash
SuperH	Sega Dreamcast	200 MHz	16 MB	CD-ROM (見內文)
ARM	Compaq iPAQ 3600	206 MHz	32 MB	16 MB flash
x86	Kontron Teknor VIPer 806	100 MHz	40 MB	32 MB CompactFlash
x86	COTS ^[註 a] Pentium	100 MHz	8 MB	32 MB DiskOnChip

註 a 現成的商品。

除了可以在嵌入式系統上執行，Linux 還可以在各種的工作站上運作。本書將會用到的主機如表 P-2 所示。儘管 Apple PowerBook 是本書主要的開發主機，但是在建立基於 x86 的嵌入式目標板時，還是會用到 x86 主機，因為有些軟體元件無法在非 x86 的主機上為 x86 目標板進行交叉編譯。舉例來說，GRUB 和 LILO 就必須建立在 X86 主機上。然而，可以告訴各位的是，這整本書是在執行 Yellow Dog Linux 發行套件的 PowerBook 主機上寫成的。另一個象徵的意義是，Linux 的版圖已經從電腦世界延伸到硬體這個非常不完整的世界了。

表 P-2：本書所用到的主機系統

架構	系統類型	處理器的時脈速度	RAM 的大小	儲存空間的大小
PPC	Apple PowerBook	400 MHz	128 MB	> GB 硬碟
x86	Pentium II	350 MHz	128 MB	> GB 硬碟

為了解說可以使用 Linux 的各種目標板架構，我在各章所舉的範例會刻意使用不同的目標板硬體。表 P-3 列出了各章所用到的目標板架構。儘管各章的說明基於不同的架構，不過各章所提到的命令卻可同時應用到其他架構上。例如，若某章所舉的例子中需要用到 *arm-linux-gcc* 命令（用於 ARM 的 *gcc* 編譯器），則只要將它代換成 *powerpc-linux-gcc* 命令，就可以將這個例子應用到 PPC 架構上。表 P-3 中，若某章所列出的架構不止一個，則其所探討的主要架構會列在第一個。例如，第 5 章所舉的命令範例會將重點擺在 ARM 架構上，而與 PPC 架構有關的命令只會提到幾個。

雖然表 P-3 列出了各章範例中所用到的目標板架構，但該表並未指出其所使用的開發主機，這是因為使用哪種主機進行開發並無差別。所以除非特別聲明，否則主機的架構總是不同於目標板的架構。舉例來說，雖然第 4 章所提到的是，以 PPC 主機為 x86 目標板建立工具。不過只要稍微或不須修改，相同指令就可以在 SPARC 或 S/390 上面執行。請注意，本書前幾章大部分的內容與架構無關，所以無須提供任何架構專屬的命令。

表 P-3：各章範例所用到的主要目標板架構

章編目	目標板架構
第 1 章	無
第 2 章	無
第 3 章	無
第 4 章	x86
第 5 章	ARM、PPC
第 6 章	PPC
第 7 章	x86、PPC
第 8 章	ARM
第 9 章	PPC、x86
第 10 章	ARM
第 11 章	PPC

軟體的版本

當然，嵌入式 Linux 系統所依靠的中心軟體就是 Linux 核心。本書將焦點放在 2.4 版的 Linux 核心上，尤其是編號 2.4.18 的發行版本。以 2.4 版為依據所做的變更，大概只會影響到本書前面幾章所提供的一些資訊。也就是說，新發行版本所支援的硬體大概會比第 3 章所列的還多。不過就 2.4 版來說，本書所提到的基本程序不太可能需要變更。不過當核心的發展越過 2.4 版進入新的里程碑（亦即 2.6 版）時，本書所提到的某些程序就有可能需要更新了。

此外，本書探討了 40 種以上開放原碼及自由軟體套件的組態、安裝和使用。每種套件的維護各自獨立，而且開發步調也各不相同。因為這些套件會與時俱進（change over time），有可能本書所提到之套件的版本，等到讀者閱讀本書的時候已經過時了。為了降低軟體更新對內文所造成的影響，本書會盡可能讓內文與版本無關。由於與軟體的變動無關，所以全書的結構和各章的結構不太可能需要更動。再者，本書所提到的許多套件已經問世一段時間了，所以它們不太可能會有什麼重大的變動。例如，本書所使用的 GNU development toolchain（開發工具鏈），其中各元件的安裝、設定和使用程序，相對來說，已經好多年沒什麼變動了，所以未來也不太可能有任何重大的變動。這同樣適用本書所提到的大多數軟體套件。

本書網站

本書所提到的軟體套件有許多都還在持續開發中，這可能會造成必須修正本書部份的內容。於是我自己架設了一個網站為讀者提供更新資訊以及與本書有關的鏈結：

<http://www.embeddedtux.org/>

舉例來說，你可以到本書網站下載附錄 A 所提到的工作表，有 PDF 或 OpenOffice 兩種文件格式。

本書慣例

定寬字

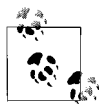
用來表示程式碼的內容或命令的輸出以及程式碼中的關鍵字。

定寬粗體字

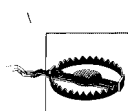
用來表示使用者所鍵入的文字。

斜體字

用來表示命令列選項、網址，以及檔案、目錄、程式和命令的名稱。



這個圖示用來表示與鄰近文字相關的秘訣、建議或通用註解。



這個圖示用來表示警告或與鄰近文字相關的警示。

建議與問題

歐萊禮公司是世界性的電腦資訊出版公司。我們永遠樂意聽到讀者對出版品的意見，包括如何讓本書可以更好的建議、指正本書的錯誤、或是讀者建議本書往後改版時，應該再加進來的其它主題。以下是本公司的聯絡資料：

美商歐萊禮股份有限公司台灣分公司

電話：(02) 2709-9669

傳真：(02) 2703-8802

網頁：<http://www.oreilly.com.tw>

電子郵件：

sales@oreilly.com.tw (業務部)
edit@oreilly.com.tw (編輯部)
bookquestion@oreilly.com.tw (書籍內容的問題)

與本書有關的線上資訊 (包括勘誤、範例程式、相關連結)：

原文書

<http://www.oreilly.com/catalog/belinuxsys/index.html>

中文書

<http://www.oreilly.com.tw/chinese/linux/belinuxsys.html>

致謝

「*E quindi uscimmo a riveder le stelle*」【註】是義大利詩人但丁在《神曲》(Divine Comedy) 第一部《地獄篇》(Inferno) 的結語。這句話可能會讓讀者誤以為我撰寫本書時沒有一點喜悅之情。但丁的這句話明確地表達了我完成你手上這本書當時的感覺。特別是我必須承認在嵌入式系統中使用 Linux 的資訊多如牛毛要加以彙整談何容易，更遑論將它們一股腦兒全都塞進一本書好告訴讀者建立嵌入式 Linux 系統的實際方法。還好，我背後有許多有能力的人願意協助我。

首先要感謝我的編輯 Andy Oram。就像協助但丁 (Dante) 遊歷靈界的維吉爾 (Virgil)，Andy 帶領著我渡過撰寫本書的每個階段。除了別的協助之外，他還耐心地糾正我非成語的成語，以確保我的文章不會辭不達意，並且鐵面無私地指出哪些章節的內容不夠深入。你現在之所以會讀到比較好的文章，這都得益於 Andy 的指教。同樣地，我要感謝 Ellen Siever，本書一開始是我跟她一起合作的。但是我們的合作關係卻在本書完成之前提早結束了，有許多點子最後之所以能夠成功放入本書，都要歸功於她建設性的回饋。

我真的非常幸運，本書能夠遇到這麼一個傑出的審閱者團隊，我非常感謝他們付出大量的時間和精神閱讀、改正以及指出本書的種種問題。他們是 Erik Andersen、Wolfgang Denk、Bill Gatliff、Russell King、Paul Kinzelman、Alessandro Rubini、David Schleaf 和 David Woodhouse。其中，我特別要感謝 Alessandro 追求盡善盡美的執著。本書接下來的內容，如果還能找到任何錯誤，毫無疑問地這都是我的問題。

●.....

註 「離開此地，只見閃耀燦爛的滿天繁星。」

撰寫關於如何在嵌入式系統中使用 Linux 的書，必須用到許多不同的硬體。這些嵌入式硬體的價錢通常很貴，我要感謝那些提供設備給我使用的公司和個人。我特別要感謝 Kontron 公司 (<http://www.kontron.de/>) 的 Stéphane Martin 提供了一張 Teknor VIPer 806 實驗版給我；DENX Software Engineering 公司 (<http://www.denx.de/>) 的 Wolfgang Denk 提供了一張 TQ components TQM860L PPC 實驗版給我；以及 Zee2 公司 (<http://www.zee2.com/>) 的 Steve Papacharalambous 和 Stuart Hughes 提供了一個 uCdim 系統給我。

身為開放原碼和自由軟體社群的使用者與貢獻者，我屢次受益於社群中其他成員所分享出來的知識與成果，這鼓舞並促成了我撰寫本書的意願。基於此，我要感謝許多人。首先，我要感謝 Michel Dagenais 教授的信任與指導，讓我有機會毫無拘束地去探索這個未經探勘的領域 (uncharted terrain)。Linux Trace Toolkit 的開發 (這是我碩士論文的一部分)，讓我與開放原碼和自由軟體社群的關係越來越密切。身為其中的一份子，我遇到了許多洞察力卓越的人，以及受到了許多協助，我非常感激他們。他們是 Jacques Gélinas、Richard Stallman、Jim Norton、Steve Papacharalambous、Stuart Hughes、Paolo Mantegazza、Pierre Cloutier、David Schleef、Wolfgang Denk、Philippe Gerum、Loic Dachary、Daniel Phillips 和 Alessandro Rubini。

最後，我一定要說，我欠 Sonia 一份情，是她異於常人的耐心，讓我得以無後顧之憂地將全部的時間投入本書的測試與撰寫。由於她的支持與關心讓本書得以順利完成。「La main invisible qui a écrit les espaces entre les lignes est la sienne et je lui en suis profondément reconnaissant.」【註】

註 「這本書的字裡行間裡充滿了你的倩影，感謝妳為我所做的一切。」