



Chapter 1

Unix、Linux、Novell SUSE LINUX

章節內容：

- Unix的歷史
- Linux的由來
- 百家爭鳴的Linux Distribution
- Novell SUSE
- 企業級SLES 9的優勢

Novell SUSE LINUX Enterprise Server 9 管理手冊

1-1 Unix的歷史

不管是IBM的AIX、Sun的Solaris、HP的HP-UX、還是近來炙手可熱的Linux，都是源自1969年AT&T貝爾實驗室（Bell Labs）所開發出來的「Unix」（Linux kernel的開發者—Linus Torvalds也誕生於這一年）！想對Linux的沿革有進一步的瞭解，不得不先來談談Unix的歷史，讀者若有興趣，也可以參閱Perter Salus所著之『A Quarter Century of Unix』一書，此書可說是Unix作業系統公認的標準史。

1965~1969年—失敗的MULTICS研究計劃

回顧Unix歷史，MULTICS（MULTiplexed Information and Computing Service）計劃可說是其發展的起源。時光回溯到1965年，當時大部份電腦都是採用批次處理（Batch Processing）的方式；批次處理的概念就像是在洗衣服，將衣服（Jobs）累積到一定的數量後才丟到洗衣機（CPU）處理，平時洗衣機（CPU）是閒置不用的。

NOTE

MULTiplexed有多人多工的涵義。

為了提升CPU的效能，美國電話及電報公司（American Telephone and Telegraph Inc.；AT&T）、通用電器公司（General Eletrics；G.E.）及麻省理工學院（Massachusetts Institute of Technology；MIT）計畫合作開發一個多用途（General-Purpose）、分時（Time-Sharing）及多使用者（Multi-User）的作業系統。當時MIT設計的CTSS（Compatible Time-Sharing System）在分時系統的發展已相當成功；所以「MULTICS（MULTiplexed Information and Computing Service）」就是計畫由CTSS擴展，建置在G.E.大型電腦「GE-645」上，以發展一套更新更好的分時作業系統。

NOTE

MULTICS 計畫所建置的機器，有些文獻寫「GE-645」，有些則寫「GE-635」；其實是G.E.為了這個計劃，將「GE-635」的硬體做了部份調整，因為與原有的「GE-635」有所不同，故稱為「GE-645」。

這裏筆者先打個岔，什麼是作業系統呢？作業系統是一個介於「電腦硬體」和「應用程式」間的程式（圖1-1），其最重要的任務就是提供應用程式的使用介面，並能管理各項硬體資源，使電腦能運行的更有效率。

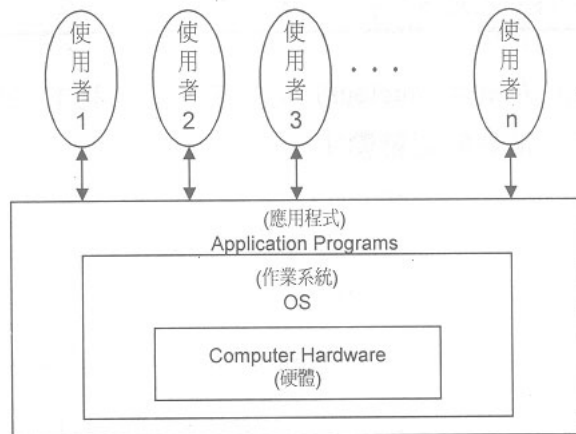


圖1-1：作業系統與電腦硬體、應用程式關係圖

依特性來看，作業系統可分為單工與多工兩類。單工的作業系統，如DOS，通常是等一個應用程式執行完後才會執行另一個程式，因此應用程式可以無限制地使用系統的CPU時間、記憶體和I/O...等資源。這個應用程式甚至可以佔據系統，不再將主導權交回作業系統，單工的作業系統只須提供系統呼叫便可運作。

Novell SUSE LINUX Enterprise Server 9 管理手冊

而多工的作業系統，如MULTICS、Unix和Linux，通常會允許多個應用程式同時進行，共用CPU時間和記憶體，因此只提供系統呼叫的服務是不夠的，還必須將硬體資源做合理的分配與管理，使這些同時執行的應用程式不會互相干擾（例如A程式的資料不能寫入B程式佔用的記憶體），才能確保這些應用程式順利地完成任務。

由於MULTICS有不少設計在當時是極新穎且頗富創意的，因此吸引了不少研究機構的注意。當時仍和AT&T同屬一家公司的貝爾實驗室（Bell Labs）更在60年代末期參與此計劃；Unix之父Ken Thompson也剛好由加州柏克萊大學（University of California at Berkeley；UCB）換到貝爾實驗室（Bell Labs），成為MULTICS研究小組的一員。

不過MULTICS的發展進度緩慢，原本預定兩年就應該有成果，但卻遲遲不見有進展；所以到了1969年2月，貝爾實驗室（Bell Labs）的有力人士便決定退出MULTICS計劃。

雖然貝爾實驗室（Bell Labs）退出，但MULTICS計劃仍持續發展著。在1972年Honeywell併購G.E.的電腦部門後，MULTICS更進入了商業市場，但始終不太成功。MULTICS比較盛行的八十年代，約有75至100台價值數百萬美元的大型電腦使用此作業系統！然而1977年，MIT也退出MULTICS的開發。八十年代中期，Honeywell把電腦事業賣給 Bull 後，MULTICS的發展也終於在1988年劃上休止符。

🌀 1969年八月－妙不可言的人月（Man-Month）

Unix之父Ken Thompson曾在MULTICS計劃中，為「GE-645」寫了個名為「太空旅行（Space Travel）」的遊戲程式；當貝爾實驗室（Bell Labs）退出MULTICS計劃後，Ken Thompson打算找一台機器把此遊戲移植過去，於是便和Dennis Ritchie向貝爾實驗室（Bell Labs）申請一台「DEC-10」的機器，以建立一台交談式、多使用者的分時系統，可惜當時申請並沒有通過。最後，他們就用一台已棄置在旁的「Digital PDP-7」迷你電腦（minicomputer）進行他們的計劃。這台機器就是Unix歷史中赫赫有名的「角落中乏人問津的PDP-7（Little-

used PDP-7 in a corner)」。

「PDP-7」是DEC (Digital Equipment Corporation) 在1964年生產的迷你電腦，其所執行的作業系統對於Ken Thompson而言是十分簡陋的，於是Ken Thompson就以他在MULTICS計劃中學到的技術和經驗，為這台過時的電腦撰寫一套新的作業系統。這套作業系統有很多構想是來自MULTICS，包括樹狀結構 (tree-structured) 的檔案系統、命令解譯器 (command Interpreter) 及對週邊設備 (device) 的存取控制方式。

最初，Ken Thompson並不是直接在「PDP-7」上撰寫新的作業系統，而是先在「GE-645」上使用GEMAP組譯器 (Assembler) 的巨集 (Macros) 編寫程式，再經由一後置處理器 (post-processor) 產生可供「PDP-7」讀取的資料。這樣一來一回，由「GE-645」到「PDP-7」，直到產生具備核心 (Kernel)、編輯器 (Editor)、組譯器 (Assembler)、Shell及一些公用程式 (如rm、cat、及cp...等) 的雛型，當整個系統可以自給自足後，所有工作才移到「PDP-7」繼續開發。

讀者可能會好奇Ken Thompson花了多少時間完成這個新的作業系統？答案是「一個月」！1969年8月，Ken Thompson以短短四個星期的時間，在DEC的「PDP-7」完成了Unix作業系統的原型。

『A Quarter Century of Unix』的作者Perter Salus說：「這是他畢生所見到，最偉大的程式設計大師的一個月，令人嘆為觀止」！筆者覺得Ken Thompson開發Unix作業系統時，有幾個概念影響後世頗深：

- 所有架構 (甚至硬體本身) 均用一個一個的檔案表示
【Everything (including hardware) is a file】
- 以文字形式儲存設定檔資料
【Configuration data stored in text】

Novell SUSE LINUX Enterprise Server 9 管理手冊

- 程式盡量朝向小而單一的目標設計
【Small, single-purpose program】
- 盡量避免令人困惑的使用者介面
【Avoid captive user interfaces】
- 可將幾個程式連結起來，處理大而複雜的工作
【Ability to chain program together to perform complex tasks】

受限於「PDP-7」機器的效能，這套作業系統只能支援兩個使用者，實在稱不上多使用者（中國老祖宗不是說過「三人為眾」）；當時和Ken Thompson一起共事的Brian W. Kernighan為了玩弄「MUITICS」這個字及強調還稱不上多使用者作業系統，便將其命名為UNICS（UNiplexed Information and Computing System），後來取其諧音，就稱其為「Unix」。

提起Unix的名字，很多人都疑惑正確寫法是全部大寫的「UNIX」，還是僅開頭字元大寫的「Unix」？Dennis Ritchie如是說：「UNIX這個寫法源自1974年CACM的文件『The UNIX Time-Sharing System』，當時這群學者剛剛取得新的typesetter以開發了troff；這些人對大寫字款著了迷，導致了『UNIX』的產生」。即使後來Dennis Ritchie多次以「Unix」不是任何句子的縮寫為由，想在幾份貝爾實驗室（Bell Labs）的文件用回「Unix」，然而「UNIX」的用法仍然相當普及。而後來全部大寫的「UNIX」也變為註冊商標。

1971年—Unix第一版

由於「PDP-7」的計算能力有限，也不太可能提供平行處理的服務，於是Ken Thompson和Dennis Ritchie再次提出申請，以研究文書處理系統的名義請購一台「PDP-11/20」。由於「PDP-11」的價格只是「DEC-10」的鳳毛麟角，而且申請的目的十分具體，AT&T的管理當局便寬宏大量地為他們購買了「PDP-11/20」型的機器。

由於當時整個系統全是以硬體相依性甚高的組合語言所編寫，要將Unix移植到「PDP-11/20」上，可不是件輕而易舉的事！他們在「PDP-11/20」將Unix加上一個編輯器，開發出一套文書處理系統—就是現在Unix作業系統中文書處理系統（`nroff/troff`）的前身。

中間有個小插曲，因為「PDP-11/20」的磁碟比主機足足慢了三個月才運到實驗室，就在等候磁碟的空窗期，Ken Thompson用「PDP-11」的組合語言（Assembly Language）重寫了整個Unix核心及基本的命令。當時那台「PDP-11/20」只有24KB實體記憶體，而這個最早期的「PDP-11」版Unix只佔用了12KB，其餘的記憶體就被用作RAM Disk和用戶的程式執行（筆者不由得想說聲：「Ken真是太神了」！）。在毫無記憶體保護（Memory Protection）和僅有500KB磁碟空間的情況下，Ken Thompson領軍的Unix小組繼續Unix的發展工作。

此時，貝爾實驗室（Bell Labs）的專利局正在尋找一個系統作為處理文件的工具；Ken Thompson所開發的文書處理系統恰好符合需求，貝爾實驗室（Bell Labs）的專利局就成了Unix的第一家商業用戶。

1971年11月，Unix小組終於將Unix第一版的說明手冊（Manual）定稿，該系統的手冊被標示為「**First Edition**」。往後由貝爾實驗室（Bell Labs）發展的Unix版本，都是以同時出版的說明文件版本號碼為準；例如Unix Version 1 (V1)其實就是指Unix第一版說明書所載的Unix。而在貝爾實驗室（Bell Labs）內部的Unix的版本一直都稱呼為「nth Edition Unix」，不是外界的「Unix Version n」。

Unix第一版已有了`ed`、`fork()`、`roff`（`troff`的前身）及基本的檔案系統...等功能，並被用作處理專利文件的工具。而`pipe()`...等功能則在Unix Version 2中首次出現。

● 1973年2月—以C語言開發的Unix第三版

在開發Unix第一版時，由於「PDP-7」和「PDP-11」的組合語言差異甚大，令移植工作非常困難。Ken Thompson與Dennis Ritchie覺得，使用組合語言開發作業系統，雖然能讓機器發揮較高的效能，但如果使用與機器相依性較低的高

Novell SUSE LINUX Enterprise Server 9 管理手冊

階語言 (High-level Language) 來撰寫，若能保有不錯的效能，那麼Unix就能移植到各種機器上，再也用不著為特定的機器寫軟體，而且往後的維護工作將會變得簡單許多！在那個完全以組合語言來開發作業系統的年代，Ken Thompson與Dennis Ritchie的想法算是相當瘋狂！

Ken Thompson首先嘗試使用FORTRAN，不過沒有成功。後來用一個叫BCPL (Basic CPL) 的語言開發。BCPL是Martin Richards在1967年於劍橋設計的；CPL (Combined Programming Language) 則是1963年倫敦大學和劍橋大學的合作項目，CPL的設計頗受1960年開發的 Algol60所影響。

Ken Thompson與Dennis Ritchie在使用BCPL期間，整理BCPL許多的功能，產生另一個新的程式語言—B語言。Unix第一版中就有少量公用程式是用B語言來撰寫。後來Dennis Ritchie覺得B語言有缺乏資料型態 (Data Type) 支援..等缺點，便著手改良B語言，誕生了大名鼎鼎的C語言。

1973年，對Unix而言是跨時代的一年，Ken Thompson和Dennis Ritchie成功地利用C語言重寫了Unix核心，此為Unix第三版。Unix至此在修改上更為便利，硬體的可移植性也較高；種種優點，奠定了Unix普及化的基礎。

NOTE

Ken Thompson與Dennis Ritchie是唯一兩位獲得Turing Award (電腦界的諾貝爾獎) 的工程師，其他獲獎人都是學者。

1973年11月—Unix第四版

Unix第四版完成後，有幾家大學和貝爾實驗室 (Bell Labs) 簽定協議，獲得了第四版的拷貝。在還沒有軟體授權觀念的時代，協議的內容主要是確保原始碼不被洩露。Ken Thompson自己錄製磁帶，不收任何費用，第一卷磁帶就由紐約的哥倫比亞大學 (Columbia University) 獲得。

1974年，Ken Thompson與Dennis Ritchie在『Communications of the ACM』上發表介紹Unix系統的論文。那時，『Communications of the ACM』是計算機科學的主要刊物，那篇文章在學術界引起了廣泛的興趣。所以，Unix第五版就以「僅用於教育目的」的協議，提供給各大學作為教學之用，成為當時作業系統課程中的範例教材。此時，Unix的使用者愈來愈多，甚至逐漸取代原來DEC公司在「PDP-11」機器上的作業系統。這種推廣播種的方式，對日後Unix幾乎成為主流有著極大的影響，也為Unix的發展注入不少新生的活力。

1976年—走出貝爾實驗室的Unix第六版

1976年的第六版，更是Unix發展史上的重要里程碑，也是Unix在貝爾實驗室外（尤其是大學中）第一個廣為流傳的版本。在這之前，Unix只存在於AT&T內部及少數大學；1976年起，大學內使用免付權利金，Unix首次流傳到AT&T以外的地方。Unix也開始走出學術界的象牙塔，透過授權（License）的方式，進入商業市場。

1978年—System IV和4.x BSD分庭抗禮

1978年，對Unix而言是革命性的一年；因為學術界的老大柏克萊大學（UC Berkeley），推出了一份以第六版為基礎，加上一些改進和新功能而成的Unix。這就是著名的「1 BSD（1st Berkeley Software Distribution）」，開創了Unix的另一個分支：BSD系列。

同時期，AT&T成立USG（Unix Support Group），將Unix變成商業化的產品。USG對外發表的第一版本是System III，下一個版本為System IV。從此，BSD的Unix便和AT&T的Unix分庭抗禮，Unix就分為System IV和4.x BSD這兩大主流，各自蓬勃發展。到了八十年代末才開始有整合的動作，如Sun Solaris OS及POSIX。

Novell SUSE LINUX Enterprise Server 9 管理手冊

NOTE

POSIX (Portable Operation System Interface)

由於有太多廠商各自發展多樣化的Unix系統，導致各Unix系統彼此互不相容；因此美國NIST / CSL便在1988年，召集各家Unix廠商、使用者與Unix組織，共同制定Portable作業系統的標準介面「IEEE 1003.1」，也就是POSIX。POSIX的全名是Portable Operation System Interface，主要的內容是對程式的撰寫制定一套標準的格式，例如針對：C programming interface、Library的system calls、Files、Process和Terminal I/O...等的程式寫法予以規範，使遵守此規範的應用程式原始碼具備portability的能力，並在Unix產品做測驗和認證，使各Unix平台規格相容統一。程式設計師只要遵守POSIX規定寫程式，那麼不管在那個Unix平台所撰寫的原始碼，都可以很容易地移植到其他的Unix平台上。

分裂的1979年—最後一個真正的Unix

1979年發佈的第七版，在許多Unix玩家的心目中是「最後一個真正的Unix」，也是「空前絕後的一個Unix」。這個版本包括一個完整的K&R C語言和Bourne Shell；而令人驚訝的是，Unix第七版的kernel 竟只有40 Kbytes！

後來Unix第七版又被移植到VAX上，稱為「32V」。由於Unix輕薄短小，許多人加入Unix的研發工作，如伊利諾大學 (University of Illinois)、哈佛大學 (Harvard University)、DEC...等，其中以柏克萊大學 (UC Berkeley) 的成果影響最深遠，柏克萊大學 (UC Berkeley) 的Bill Joy和Ozalp Baboglu以「32V」為基礎，加入虛擬記憶體、需求分頁及分頁替換等重要功能後，發表3 BSD Unix。因為它具有大型的記憶體管理機制，因而獲得DARPA (美國國防部高等防衛研究所) 的支持。

3BSD Unix除了增加32位元的虛擬記憶體功能之外，也加入許多公用程式 (如：vi 編輯器...等)，並把現在必備的操作介面C Shell加至系統中。BSD因此得以迅速的發展，而成為另一種風格的Unix。到了4.2 BSD時，它已能支援

LAN（如Ethernet和Token Ring）及WAN（如NSFNET）。

可惜AT&T最後竟使出殺手絕招，一狀告到法院，說BSD剽竊AT&T的原始碼，而終結了頗受好評的BSD系統。而AT&T本身則因反拖拉斯法被拆解成幾個子公司，使其可以正大光明的為Unix商品化，全心全意的發展整個作業系統。

由於Unix是由C語言寫成的，只要稍微修改原始碼，便可在不同的硬體架構上面執行。因此，很多商業公司及學術機構均加入這個作業系統的研發，各個不同版本的Unix也開始蓬勃發展，例如AT&T的System V、加州大學的BSD版、IBM的AIX...等。

● 1984年－麻省理工學院（MIT）的雅典娜計畫 （Project Athena）

有鑑於圖形使用者介面（Graphical User Interface；GUI）的需求日益加重，1984年，麻省理工學院（MIT）發展一個可以讓視窗介面適應多平台環境的雅典娜計畫（Project Athena），並與其他協力廠商發表了X Window System。1988年，非營利性質的XFree86組織成立，所謂的XFree86其實是X Window System與Free及X86系統架構的整合名稱。而這個XFree86的GUI介面，更在1994年Linux的核心1.0版釋出時，整合於Linux作業系統中。

● 1984年－GNU（GNU is Not Unix）計畫：未完成的kernel

由於AT&T的態度，讓Unix在商業應用及改良上出現許多問題，而且不能自由修改及散佈更好的Unix作業系統。這些商業軟體的種種限制，讓許多Unix喜好者及軟體開發者感到相當的憂心；著名的Richard M. Stallman先生正是其中一名。他認為，Unix是一個相當好的作業系統，如果大家都能夠將自己所學貢獻出來，那麼這個系統將會更加的優異！

Stallman強調，軟體的開發應該要有Open Source的概念，讓大家都能夠分享別人的結果，並將自己的心得貢獻出來，讓軟體趨向更好、更實用。藉由Open Source的想法，開發的程式將有很多人可以幫忙檢驗，使得Unix社群更加

Novell SUSE LINUX Enterprise Server 9 管理手冊

的茁壯。

為了自己的理想，Stallman在1984年實際創立GNU（GNU is Not Unix）。GNU計劃開發一套與Unix向上兼容，名為「GNU」的自由軟體系統。並在1985年成立自由軟體基金會（Free Software Foundation；FSF），創作了許多「自由軟體」供大眾使用。

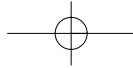
Stallman開發GNU作業系統相關組件時，他最主要的關切點，便是要確保他和同伴所開發的程式碼可以自由散佈，所以他起草一份保護自由軟體的版權聲明文件—「GNU／GPL（GNU General Public License）」，或是更簡短的稱為GPL。

GPL是一種旨在保護而不在限制使用者權利的軟體授權方式。與一般軟體授權方式最大的不同點，在於使用GPL當作軟體的授權條件時，必須允許使用者擁有自由修改軟體的權利，同時也可允許使用者自由散佈修改後（或沒修改過）的軟體，但使用者必須要繼續遵守GPL授權。「GNU／GPL（GNU General Public License）」全文可參考<http://www.gnu.org/copyleft/gpl.html>，中譯版本的網頁請見http://www.slat.org/project/legal/GNU_GPL_Chinese。

GNU程式計畫開發出不少驚人的軟體，尤其是GCC。什麼是GCC？就是GNU的C語言編譯器（GNU C Compiler；GCC）。GCC語言編譯器對推廣「GNU」計劃，著實有莫大的助益。

另外兩個大計劃，當屬Shell程式及C語言函式庫（C library）的創作了！Shell程式提供作業系統基本的介面，Stallman曾說：「要鍵入指令，首先要有Shell程式」。GNU的Shell程式叫bash，這是Bourne Again Shell的縮寫，同樣也是對Unix開了個小玩笑，因為Unix的Shell程式叫Bourne Shell。

C語言函式庫（C library）則是讓一大堆程式可以呼叫及共用的程式碼，它可以大幅縮小使用者開發程式的大小及時間。C語言函式庫（C library）、bash還有許多Unix作業系統的元件，都在1990年左右完成，但仍少了最重要的程式—kernel（核心）。Kernel是作業系統最底層的東西，掌管整個硬體資源的工作狀態，除了控制系統硬體的資源，也負責管理電腦軟體程式的執行，可說是介



於硬體和使用者間的介面，提供使用者使用系統所需的服務。硬體、kernel、Shell、C 語言編譯器彼此間的關係可參考圖1-2。當年因為種種原因，Stallman 一直未完成kernel的開發，最後由Linus Torvalds所開發的kernel—Linux彌補了這個缺憾。

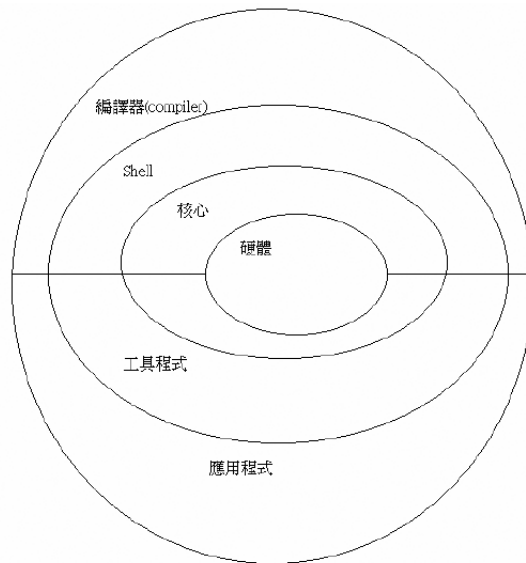


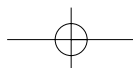
圖1-2：作業系統簡圖

1991年—UNIX商標數度易主，Linux即將掘起！

1991年4月，AT&T成立子公司Unix System Laboratories (USL)，自此，UNIX這個商標便由USL所擁有。

1993年，網路巨人Novell出奇不意的買下USL，並破天荒的把UNIX商標讓渡給X/Open，Unix的未來正如他的過去，充滿了璀璨而令人目不暇給的光芒。至於Unix未來的統一大業，誰也不敢預言，畢竟有了過去的紛擾與風雨，也才有今天成熟又好用的Unix。

其實，Unix的故事仍舊延續著...許多網站也為這段歷史留下記錄。圖1-3列



Novell SUSE LINUX Enterprise Server 9 管理手冊

出一個詳細記錄Unix歷史的網站 (<http://www.levenez.com/unix/>)，這個網站忠實記載著1969~2005年Unix發展的大事，而且還有PDF檔案可供下載。網站的首頁陳列每個時期Unix的歷史，也代表著無數工程師的心血與努力。

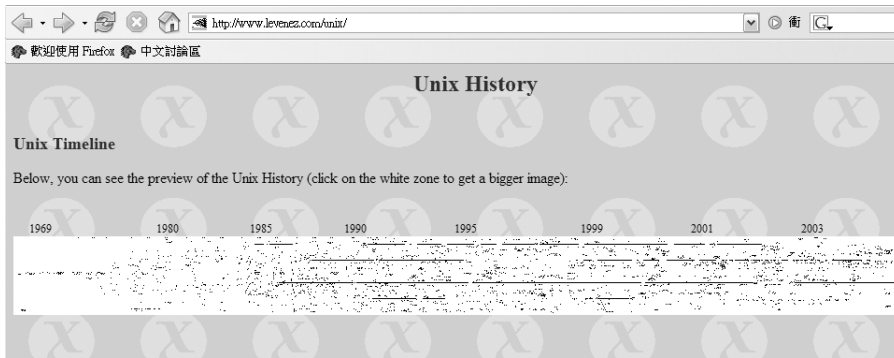


圖1-3：介紹Unix歷史的網站 (<http://www.levenez.com/unix/>)

1978 1BSD	1982 HP-UX	1977 RT
1979 2BSD	2000 HP-UX 11i	1994 SCO UNIX
1980 3BSD	1993 HP-UX BLS	2002 SCO UnixWare 7
1980 4BSD	1988 IBM AOS	1984 SCO Xenix
1994 4.4BSD Lite 1	1985 IBM IX/370	1987 SCO Xenix System V/386
1995 4.4BSD Lite 2	1985 Interactive 386fix	2001 Security-Enhanced Linux
1992 386 BSD	1978 Interactive IS	2004 Silver OS
1986 A/UX	1988 IRIX	1983 Sinox
1989 Acom RISC iX	1991 Linux	1995 Sinox ReliantUnix
1988 Acom RISC Unix	1994 Lites	1990 Solaris 1
1990 AIX	1977 LSK	1992 Solaris 2
2000 AIX SL	1999 Mac OS X	1982 SPIX
1989 AIX PS/2	1999 Mac OS X Server	1982 SunOS
1990 AIX/370	1985 Mach	2004 Triance OS
1989 AIX/6000	1974 MERT	1999 Tru64 Unix
1991 AIX/ESA	2002 MicroBSD	1995 Trusted IRIX/B
1986 AIX/RT	1977 Mini Unix	1998 Trusted Solaris
1990 AMiX	1984 Minix	1991 Trusted Xenix
1995 AOS Lite	2000 Minix-VMD	1977 TS
1992 AOS Reno	1985 MIPS OS	1980 UCLA Locus
1994 ArchBSD	2002 MirBSD	1979 UCLA Secure Unix
1991 ASV	1996 Mk Linux	1988 Ultrix
1989 Aten Unix	1998 Monterey	1984 Ultrix 32M
1989 BOS	1988 moreBSD	1982 Ultrix-11
1979 BRL Unix	1983 mXinu	1984 Unicos
1988 BSD Net/1	1993 MVS/ESA OpenEdition	1996 Unicos/mk
1991 BSD Net/2	1993 NetBSD	1993 Unicos-maz
1991 BSD/386	1988 NeXTSTEP	1969 UNICS
1992 BSD/OS	1987 NonStop-UX	1979 UNIX 32V
1978 CB Unix	1994 Open Desktop	1991 UNIX Interactive
1986 Chorus	2001 Open UNIX 8	1981 UNIX System III
1988 Chorus/MiX	1995 OpenBSD	1982 UNIX System IV
1983 Coherent	1995 OpenServer 5	1983 UNIX System V
1987 CTIX	1996 OPENSTEP	1984 UNIX System V Release 2
1999 Darwin	1996 OS/390 OpenEdition	1986 UNIX System V Release 3
2000 Debian GNU/Hurd	1997 OS/390 Unix	1988 UNIX System V Release 4
1995 DEC OSF/1 ACP	1990 OSF/1	1985 UNIX System V/286
1995 Digital Unix	1982 PC/IX	1986 UNIX System V/386
2003 DragonFly BSD	1986 Plan 9	1971 UNIX Time-Sharing System
1984 Dyaix	1977 PWB	1993 UnixWare
1993 Dyaix/ptx	1974 PWB/UNIX	1998 UnixWare 7
2003 ekkoBSD	1984 QNX	1976 UNSW
1977 Eunice	2001 QNX RTOS	1977 USG
2004 FireFly BSD	1996 QNX/Neutrino	1982 Venix
1993 FreeBSD	1981 QUNIX	1980 Xenix OS
1986 GNU	1997 ReliantUnix	1984 Xinu
2001 GNU-Darwin	1997 Rhapsody	1998 xMach
1987 HPBSD	1991 RISC iX	2001 z/OS Unix System Services

圖1-4：各種Unix-Like作業系統發展的年代

NOTE

Linux、IBM AIX、Sun Solaris、FreeBSD、BSD、Sun Unix、SCO Unix、HP Unix...等，都是由同一個祖先「Unix」衍生而來。所以這些作業系統稱為Unix-Like的作業系統。

回顧Unix歷史，最主要有兩大流派：那就是AT&T的System V與BSD (Berkeley Software Distribution)；SVR4則是兩大流派融合後的產物。底下列出這兩大派系的主要版本與版本特色。

System V的主要版本

- System III (1981)：AT&T 第一個拿來營利的Unix。
 - FIFOs (named pipes)。
- System V (1983)
 - IPC package (shm, msg, sem)。
- SVR2 (1984)
 - Shell函數 (sh)。
 - SVID (System V Interface Definition)。
- SVR3 (1986)
 - STREAMS、poll()、TLI (網路軟體)。
 - RFS。
 - 共用程式庫 (shared librarys)。
 - SVID 2。
 - demand paging (如果硬體有支援的話)。
- SVR3.2 (1987)
 - 併入Xenix (Intel 80386)。
 - 網路。

Novell SUSE LINUX Enterprise Server 9 管理手冊

- SVR4 (1988) : 融合了System V、BSD、SunOS，是各種Unix的主流。
 - 取自SVR3者：系統管理、Terminal介面、印表機、RFS、STREAMS、uucp。
 - 取自BSD者：FFS、TCP/IP、sockets、select()、csh。
 - 取自SunOS者：NFS、OpenLook GUI、X11/NeWS、共用程式庫、具有記憶體對映檔案的虛擬記憶體子系統 (virtual memory subsystem with memory-mapped files)。
 - ksh。
 - ANSIC。
 - 國際化 (8-bit clean)。
 - ABI (Application Binary Interface — routines instead of traps)。
 - POSIX、X/Open、SVID3。
- SVR4.1 (1992)
 - 非同步 I/O (取自SunOS)。
- SVR4.2 (1992) : based on SVR4.1ES
 - Veritas FS、ACLs。

BSD的主要版本

BSD (Berkeley Software Distribution) 多半採用在VAX、RISC...等各式工作站。與System V比起來，BSD的變動比較快，而且學術研究的味道比較濃厚。許多對 Unix的加強改進都是由BSD先做出來的，Unix之所以能夠流行，BSD居功甚偉！

- 2.xBSD (1979) : 供PDP-11使用。
 - csh。
- 3BSD (1980)
 - 虛擬記憶體。

- 4.0BSD (1980)
 - termcap、curses。
 - vi。
- 4.1BSD (1981)：後來AT&T CRG皆以此為版本。
 - job控制。
 - automatic kernel config。
 - vfork()。
- 4.2BSD (1983)
 - TCP/IP、sockets、ethernet。
 - UFS：長檔名，symbolic links。
 - 新的reliable signals (SVR3採用了4.1的reliable signals)。
 - select()。
- 4.3BSD (1986)
- 4.3 Tahoe (1988)：4.3BSD附加對Tahoe (一款32位元的超級迷你電腦)的支援。
 - Fat FFS。
 - 新的TCP演算法。
- 4.3 Reno (1990) for VAX、Tahoe、HP 9000/300
 - 大部份的P1003.1。
 - NFS (from Sun)。
 - MFS (記憶體檔案系統)。
 - OSI：TP4、CLNP、ISODE's FTAM、VT and X.500、SLIP。
 - Kerberos。
- Net1與Net2 (June 1991)
 - 4.4BSD (alpha June 1992) for HP 9000/300、Sparc、386、DEC。

而與System V針鋒相對的Open Software Foundation，於1991年底推出了OSF/1；OSF/1需要SVR2授權，符合SVID 2、SVID。

1-2 Linux的由來

由於版權問題，Unix的原始碼不再適用於教學，1987年Andrew Tanenbaum遂寫了MINIX作為教學的工具；MINIX的意思為 mini-Unix，它是一個簡化的作業系統，適合入門者學習。因為簡單，開始時獲得眾人的青睞，但好景不長，主要還是因為過於簡單的設計反而不切實用。

1991年，芬蘭赫爾辛基大學的學生Linus Torvalds，由於在使用MINIX時對其提供的功能不甚滿意，自行發展作業系統（那時，Linus Torvalds才25歲）。Linus Torvalds利用USENET NEWS對全世界的MINIX使用者發表文章，大意是聲明自己正在發展一套「免費」的作業系統，專供386或486 AT相容電腦使用，希望貢獻給所有MINIX的同好，並強調這套系統雖然有一點像MINIX，但是完全沒有使用MINIX的原始碼，所以沒有版權上的問題。這就是Linux的開端—kernel 0.01（圖1-5）。

"Hello everybody... I'm doing a (free) operating system (just a hobby, won't be big and professional...)."

Linus Torvalds, creator of Linux, from the first Internet announcement on August 25, 1991. Even he initially underestimated its potential.



圖1-5：Linus Torvalds的Linux宣言

Linus所開發的這個「hobby」是基於Unix系統發展出來的，並且使其可以在Intel主導的x86的系統下執行（當初的個人電腦系統為386與486架構）。原本這套系統命名為「freax」，後來被稱為「Linus's Unix」，簡稱為Linux。Linus將這套系統放到全芬蘭最大的FTP伺服器，建了一個Linux的子目錄，允許人們自

由使用 (under GNU Public License)。同年10月，Linus正式推出 Linux 0.02 版，並在Internet中廣邀高手加入Linux的發展工作，更歡迎把MINIX上免費程式移植到Linux中，因此研究成果很快就散佈到世界各個角落。

之後的發展更是快速，幾乎一兩個禮拜就有新版或修正版的出現，時至今日，kernel的版本已經出到2.6X。Linux的發展不像傳統的軟體工程，它完全是透過網路，集合世界各地的高手而成的一套作業系統，在這裡我們也可以見識到網路快速傳播的威力。

至於Linux為什麼可以如此「火」，筆者將其歸納為下列幾點：

● Linux是個完全免費的Unix

Linux是個完全免費的Unix！藉由許多人的修改與加強，如今Linux除了可以在原先設計的Intel x86上執行，它也被移植到 Alpha、Sun Sparc、Motorola 68K、MIPS、PowerPC...等平台上。

● Linux雖是Unix的翻版，但並未用Unix的原始碼

雖然Linux擷取許多Unix的構想，並實作Unix的API（像是POSIX和Single Unix Specification所定義的標準），但它卻不像其他的Unix系統直接修改Unix的原始碼，所以不受當年AT&T的版權所限。它採用和Unix系統不同的實作方式，但卻不背離Unix的整體設計目標，也不破壞其應用介面。

由於Linux提供和Unix類似的介面，因此，在資源的抽象化及資源分享的形式上就必須和Unix相同，而實作這些介面時所使用的資料結構及演算法，則和Unix有所不同。

● Linux是多人多工的作業系統

Linux可以同時允許多人上線工作，並且合理地分配資源給不同的使用者。Linux也是一個多工的作業系統，許多程式可以一次讀到記憶體內，作業系統執行某個程式一段時間後，就會切換去執行另一個程式。所以記憶體是被「空間」分割成數個區塊，每個區塊稱為memory partition，內含一個隨時可以執行的程

Novell SUSE LINUX Enterprise Server 9 管理手冊

式。CPU 則是被「時間」分割，每段時間執行一個在記憶區塊內的程式，時間到了便切換到另一支程式。在Intel x86平台上，CPU每秒約切換100次，在Alpha 平台則切換約1024次；這些切換次數可依需要而修改。大致而言，降低切換次數會使I/O（如鍵盤、滑鼠...等）的反應變慢，但應用程式執行的效率會提高；相反地，如果提高切換次數，則會使程式的執行變慢，因為更多的CPU 時間被用來切換程式。

● 集成式的核心（monolithic kernel）是最適合的技術

Linux 核心和傳統的Unix一樣，是屬於集成式的作業系統核心（monolithic kernel），和目前流行的微核心（microkernel）不同。它們將行程管理、記憶體管理和檔案系統全包在一起，成為一個單一的可執行檔；而週邊硬體裝置管理則另外分開，成為一組驅動程式，每一個驅動程式的目的是控制某一類型的硬體裝置（例如控制軟碟機）。這種設計是為了降低核心更動的頻率，不必為了新硬體裝置更改核心，而且驅動程式也比較好寫。但是 Linux 核心的進步非常迅速，這種設計反而不利於核心的實驗更新，為了克服這個缺點，Linux 提出模組（module）機制，它和核心的介面要比傳統 Unix的驅動程式來的有彈性，可以用來提供新功能給核心，當然也適用於驅動程式

● 硬體需求較低且運行穩定

Linux 只要一般等級的個人電腦就可以安裝；既然是Unix-Like的作業系統，當然也繼承了Unix 穩定與高效能的特點！

1-3 百家爭鳴的Linux Distribution

其實Linux就本質而言，只是作業系統的核心（kernel），但一個強大的作業系統若缺乏應用程式的搭配，就顯得用途不大了。所以一些商業公司或是非營利性的工作團隊，便將Linux核心、核心工具與相關的軟體集合起來，加入自己公司或團隊創意的系統管理模組與工具，釋出一套可以完整安裝的作業系統，這個完整的Linux作業系統，我們就稱為「Distribution」，或者是中文所謂的「發行套件」（圖1-6）。

讀者得注意，不要把Distribution的發行版本編號當成是Linux核心的版本編號，例如在SUSE LINUX Enterprise Server 9的發行版本中，數字9是Distribution的版本編號，而其Linux核心則是2.6.x的版本。

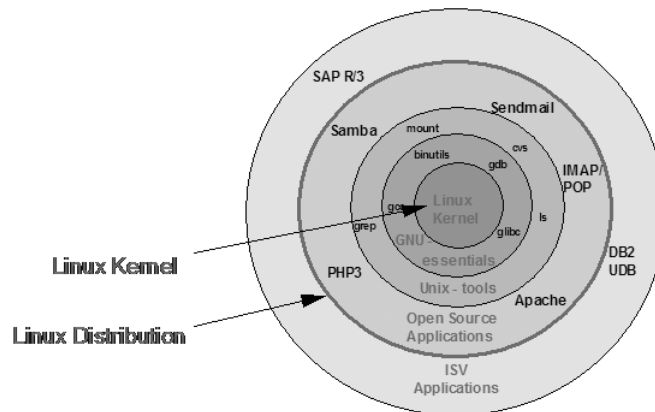


圖1-6：Linux kernel和Distribution關係圖

發展Linux的公司實在太多了，例如有名的Novell SUSE、Red Hat、OpenLinux、Mandrake及Debian...等。很多人擔心，如此一來，每個Distribution是否有很大的差異呢？其實不需要擔心！由於各個Distribution都是架構在Linux核心的基礎，來發展屬於自己公司風格的Distribution，而且大家都遵守Linux Standard Base（LSB）的規範，各個Distribution其實是差不多的！反正用到的都

Novell SUSE LINUX Enterprise Server 9 管理手冊

是Linux 核心啊！只是各個Distribution裡面所使用的套件可能並不完全相同而已。

Linux Distribution的數目非常多，若讀者想得知各個Linux Distribution的功能及熱門程度，可到<http://www.distrowatch.com/>查詢。另外，想下載多種Linux Distribution，<http://www.linuxiso.org>這個網站也提供了豐富的資源。



圖1-7：http://www.distrowatch.com中，各Distribution的熱門程度比較

1-4 Novell SUSE

● 1993年SuSE成立

SuSE原為一間德國的Linux公司，在1993年由Hubert Mantel、Burchard Steinbild、Roland Dyrhoff及Thomas Fehr四人成立。SuSE這名字是德文「Software und System Entwicklung」的簡寫，意思為「軟體及系統開發 (Software and System Development)」。SuSE最初是把Slackware的介面德文化後出售，後來他們覺得，與其不斷為Slackware除錯，不如開發自己的Distribution，於是就以Florina LaRoche的jurix distribution為基礎，開發了較易操作的設定工具YaST (Yet another Setup Tool)，藉以管理其Distribution—SuSE Linux 4.2。

除了YaST，SuSE另一個特色為KDE (K Desktop Environment) 桌面環境的使用。此外，SuSE亦開發自己的X Server—SuSE XFCom-Server；XFCom-Server與XFree86相容，並支援一些XFree86所不支援的顯示卡；而且每當XFCom的X Server穩定後，SuSE便會回饋給XFree86供各家Linux Distribution及FreeBSD使用，充分發揮Linux自由軟體的精神！

● 1997年進軍美國

很快地，SuSE Linux成為德國最暢銷的Linux Distribution，在德國Linux的市場佔有率遠遠高於其他的Unix作業系統，在歐洲其他地方亦頗受好評。1997年初，SuSE挾其在德國及歐洲等地的傲人業績，在美國加州西部的奧克蘭 (Oakland) 成立分公司，正式進軍美國。

● 2003年Novell收購SuSE Linux

2003年，Novell的CEO Jack Messman開始對公司進行大膽的Linux改造，並收購了SuSE Linux來取代過氣的NetWare作業系統；在此宗併購之外，Linux最大擁護者IBM還投下5000萬美元的投資給Novell，並於2004年1月完成合併；而SuSE Linux產品也更名為Novell SUSE LINUX。Novell是一家營收在10億美金

Novell SUSE LINUX Enterprise Server 9 管理手冊

以上的國際企業，在網路產品有著輝煌的歷史；SUSE LINUX一向是全球Linux軟體和服務的領導者，其產品的可靠度也享有相當卓越的聲譽；挾此優勢，相信Novell在今日的企業應用市場中，會有著更令人刮目相看的成就。

2004年8月推出內含kernel 2.6的SLES 9

Novell率先在2004年8月推出包含完整kernel 2.6功能的Novell SUSE LINUX Enterprise Server 9 (SLES 9)，然後在同年11月發表了Novell Linux Desktop 9桌面系統；並在2005年初推出雙作業系統核心的Open Enterprise Server (OES)；展現出以Linux作業系統為基礎，發展全系列產品線，進而成為Linux企業應用的領導者的決心與企圖心。

Novell Taiwan總經理陳學智認為，在軟體流變上，Linux和Unix其實是一脈相承，最大差異在搭載的硬體；Linux以往主要搭載Intel-based的x86機器。隨著x86機器的運算功能逐漸提升，使得它和RISC機器的差異愈來愈少，也更能負擔Unix的任務。

他同時表示：「許多企業長久以來仰賴Novell所提供的技術與解決方案，結合SUSE LINUX優異的產品性能、Novell豐富的企業軟體專業知識及其遍佈全球的技术支援，Novell將持續提供客戶穩定且安全的Linux解決方案」。

2005年SUSE LINUX佔有率急速上升

在2005年1月，安全公司SG Cowen對440位北美資訊技術管理人員進行調查，根據各網站對不同版本Linux Distribution使用率進行的比較顯示，Novell的佔有率從21%成長到了33%，足見其日益受到Linux使用者的認同。就如同Robert Frances分析師Stacey Quandt表示，SUSE LINUX已經成功地運行在很多企業資料中心的大型主機之上，也被廣泛地應用在金融服務產業的伺服器中。

1-5 企業級SLES 9的優勢

1-5-1 內含kernel 2.6

2003年12月17日，kernel 2.6問世；Novell SUSE LINUX Enterprise Server 9即採用kernel 2.6。到底kernel 2.6有那些重大的突破？下面便逐一探討。

速度最佳化

O (1) scheduler

什麼是scheduler？一個作業系統同時會有很多程序（Process）在執行，有些是使用者看得到的（如利用OpenOffice打文章），有些是使用者看不到的（如WWW Server、DNS Server...等的運作）；作業系統通常會安排這些程序（Process）輪流使用CPU一段時間，因為CPU的執行速度極快，感覺上就好像同時執行多件工作，這就是所謂的「多工（multitasking）」的概念。但是，誰來分配時間？怎樣有效分配？這就是scheduler所負責的工作（圖1-8）。

為了能夠預測系統執行時間的增長情形，我們常分析程式的執行步驟以估算執行時間的複雜度，在演算法中，通常利用O (f(n))的符號（f(n)可以是任意的函數）來表示。以scheduler分配程序（Process）所需的時間為例，如果有n件程序（Process），若是分配程序（Process）的時間就要n個單位，那這個呈線性正比關係的執行時間，可以用O (n)來表示；O (n)只是一個符號，而不是某個函數。至於O (1)代表什麼呢？O (1)表示執行時間為常數值，也就是分配程序（Process）的時間和程序（Process）數量無關，不論系統有10件、100件、甚至1000件程序（Process），分配程序（Process）所需的時間沒有任何分別，在理論上，這已是最理想的效果。

Novell SUSE LINUX Enterprise Server 9 管理手冊

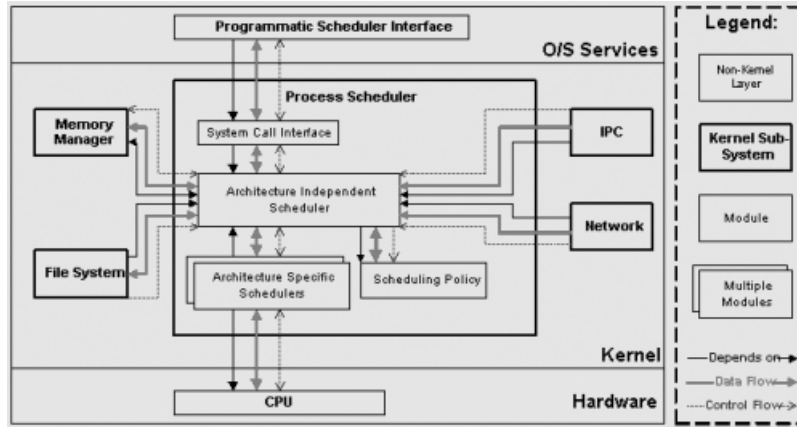


圖1-8：scheduler架構圖

Preemptive kernel

Preemptive kernel 在kernel 2.4的年代就已採用，其概念是指一個程序 (Process) 可以暫停執行，將CPU資源讓給需要優先處理的程序 (Process)。這對使用者來說是很重要的，因為系統可以依使用者需求即時作出反應，不會因為某個程序 (Process) 佔用CPU太久的時間，而令人覺得系統反應緩慢。

例如，系統有個工作要經常讀寫硬碟，當使用者開啟文書處理器時，如果沒有preemptive的功能，系統就會優先進行硬碟的讀寫動作，而使用者要編輯的內容就要慢慢等待，才會一個個出現。這種因為不斷讀寫硬碟引起的heavy load，有時還會造成系統crash。針對這種問題在kernel 2.6.0已有改善。

更強的硬體支援

更高的擴充性與支援度

- 相較kernel 2.4最多只能支援32個CPUs，kernel 2.6可支援高達255 個CPUs。
- 可存取64GB的記憶體，並不再受64bit的限制。

- 在kernel 2.4中，block device最大為2TB，kernel 2.6最大已可支援至16TB。
- 裝置分類的major numbers可至4095；minor numbers可至1,000,000。
- UID與GID增加到40億。
- PID的數目增加到10億。

支援Hyperthreading

kernel 2.6對Intel Pentium 4的讀者是一大福音，因為其支援 Hyperthreading，這是最初發佈的kernel 2.4所沒有的（kernel在2.4.17的發佈中才開始包含Hyperthreading的支援）。Hyperthreading 是甚麼？基本上，Hyperthreading可以將一個 CPU偽裝成兩個或更多的CPUs，而電腦模擬多個CPUs，可更有效率地分配工作。

支援無線wireless裝置

隨著無線技術的發展，Linux也不斷地擴充無線裝置的支援，故許多無線支援的特性就已經包含於kernel 2.4了。然而在kernel 2.6中，最令人興奮的改進就在於對藍芽（Bluetooth）技術的支援；藍芽（Bluetooth）是一種新的無線協議，主要應用在短距離的數據及語音通訊上，現在許多PDA及手機都有藍芽（Bluetooth）功能，Linux在無線裝置發展的成熟，更顯其跟的上時代潮流！

支援熱插拔（hot-plug）裝置

所謂熱插拔（hot-plug），是指系統啟動後連接上去就可以使用的裝置，例如USB。在kernel 2.6可預設支援USB 2.0裝置。

支援多種儲存裝置

在kernel 2.4時，物美價廉的SATA硬碟可說無用武之地；但在kernel 2.6便可享受SATA硬碟的好處。另外，在kernel 2.4中，IDE CD-RW是被模擬成 SCSI 裝

Novell SUSE LINUX Enterprise Server 9 管理手冊

置，各種燒錄光碟的程式會把IDE CD-RW當作是SCSI裝置來看待；kernel 2.6已重寫這部份的程式碼，不用再將IDE CD-RW模擬成SCSI裝置，而是當作真正的IDE裝置。

支援ISA 匯流排的PnP（Plug and Play）特性

對於仍在使用舊機種的讀者來說，這無疑是個好消息，因為kernel 2.6對ISA PnP的支援更完善了！原先的kernel 2.4在ISA PnP方面的設計並不完整，所以升級至kernel 2.6對於有ISA匯流排的舊機器來說絕對有好處！

發展udev

在kernel 2.4中，有一個稱為「devfs」的設計，它的目的是隨時增加或移除在「/dev」中的裝置。看過「/dev」目錄的人都知道，Linux所支援的裝置數目多得驚人，但在這逾千個裝置中，只有極少數是真正存在的；如果有辦法可以即時顯示真正存在的裝置，對使用者或各經銷商來說都是好事。「devfs」就是為這個目的而編寫。

到了kernel 2.6，這部份的發展有了變化。Kernel開發小組希望用另一種更嚴謹的方式來進行「devfs」所做的工作，於是發展了「udev」，以取代「devfs」。「udev」會被掛載（mount）在「/udev」目錄內，裡面全都是一些會即時出現或消失的裝置；它會讀取「sysfs」中提供的裝置狀態而建立或刪除不同的裝置。由於還是起步階段，目前未能真正取代可靠的「devfs」，但相信在不久的未來，「udev」的設計會更成熟穩定。

檔案系統

JFS

Novell SUSE預設是ReiserFS的檔案系統；Red Hat Linux則是使用ext3的檔案系統；IBM AIX Unix是採用先進的JFS檔案系統。雖然Linux早已嘗試將JFS加入kernel中，但穩定度仍不如AIX上的JFS；不過到kernel 2.6時，Linux上的JFS技術已逐漸完善。

NFS (Network File System)

kernel 2.6上的NFS為「NFS4」版本；而目前Linux上用的NFS大多是「NFS2」或「NFS3」。「NFS4」有甚麼特別？相較先前只能使用UDP協定的版本，「NFS4」可使用TCP協定，TCP比UDP更能保證傳送資料時不會遺失封包。另外，「NFS4」可以將資料加密，這個是「NFS2」或「NFS3」所沒有的。

筆者看到kernel 2.6內的NFS版本可是發出會心的微笑，因為筆者在專案中曾遇到以Linux做為NFS Server效能比Sun Solaris差上一大截的問題！不管筆者如何調整Linux的參數，始終無法比擬Sun Solaris的傳輸速度。原因是kernel 2.4中的NFS Server並沒有完全支援「NFS 3」，只能使用UDP協定，而且rsize及wsize最大只能到8192 bytes；然而Sun Solaris完全支援「NFS 3」，不但可使用UDP及TCP協定，且rsize及wsize最大可到32768 bytes。不過現在kernel 2.6的NFS為「NFS4」版本，效率已有長足進步，看來「日蝕現象」將愈來愈明顯。

● 電源管理

在kernel 2.4時，ACPI (Advanced Configuration and Power Interface) 的功能一直有些問題，雖說各大Distribution加入相關的修正程式，但筆記型電腦使用ACPI電源管理時始終不太穩定，甚至會造成當機或者某些功能無法使用。但在kernel 2.6，這方面的設計已有明顯的改善。

首先是支援software suspend，即將整個留在RAM的Linux系統寫入Swap partition，然後立刻關機。待要重新開機時，就將Swap partition裡的資料讀出來載入記憶體裡面，完全省去開機關機的時間。

另外則根據系統電源的狀態，自動調整CPU速度（應該稱為speedstep）。當然，這種功能要CPU有支援才可以，通常筆記型電腦的CPU才會支援。

● kernel 2.4和kernel 2.6比較表

筆者將kernel 2.4與kernel 2.6較重要的差異整理如下表，方便讀者做一比較。

Novell SUSE LINUX Enterprise Server 9 管理手冊

Kernels	2.4.0	2.4.22	2.6.x
Max # of CPUs	32	32	255
Large block device size supported (32 bit)	2TB	2TB	16TB
NUMA support	Poor	Some	Good
Preemptive Kernel	NO	NO	YES
O(1) scheduler	NO	NO	YES
CPU affinity	NO	NO	YES
Reverse mappings	NO	NO	YES
Large page support	NO	NO	YES
NFS version 4	NO	NO	YES
NFS over TCP	NO	YES	YES
CIFS	NO	NO	YES
SCTP	NO	NO	YES
IPSec	NO	NO	YES
JFS	NO	YES	YES
ResierFS	NO	YES	YES
XFS	NO	NO	YES

表1-1：kemel 2.4與kemel 2.6比較表

1-5-2 SUSE AutoBuild - one source for all platform

SUSE LINUX採用開放原始碼業界中獨到的研發與製程方法—「AutoBuild」；AutoBuild是以同一套共通程式碼為基礎（common code base）來提供廣泛硬體平台的執行碼。從桌上型電腦到大型主機，包括x86、AMD64、Intel EM64T、Intel Itanium系列、IBM pSeries、IBM zSeries，確保部署在所有硬體版本的SUSE LINUX產品都擁有相同的品質保證。

1-5-3 獨一無二的系統管理工具 - YaST

SUSE LINUX 具有獨一無二的管理功能—YaST (Yet another Setup Tool)，是最容易在企業中部署、組態與管理維護的Linux系統。其統一的圖形化管理工具，簡化了作業系統、網路服務、儲存、叢集 (cluster) 乃至應用程式的安裝與設定工作。AutoYaST可以在網路架構上實現「no touch」的自動化Linux部署開放應用程式介面 (API) 和支援通用資訊模型 (CIM) 標準，允許與協力廠商的管理解決方案相互整合。

1-5-4 EAL4+安全等級

EAL (Evaluation Assurance Level) 安全認證體系是國際公認的ISO標準 (ISO/IEC 15408)，被很多國家、政府、組織和專家採用，做為關鍵性軟體的安全衡量標準。

1-5-5 SLES 9優點總結

總結上述內容，SLES 9有下列優點：

- 全球第一個建立在高效能 Linux 2.6 核心基礎的企業級 Linux 伺服器，支援最多 512 個 CPUs、32000 個 SCSI Disks、4095 device types、超過 40 億個使用者、65535 個使用者程序。
- 最大的檔案系統容量 (per partition/file system size) 高達 16 TB。
- 具進階記憶體管理和 native POSIX threading library、Hyperthreading 功能，可以同時/平行處理更多應用系統的交易量，並有效加速回應時間；具有進階 I/O 效能規劃程式，可迅速調整應用程式效能。
- 等級化核心資源管理 (Class-based kernel resource management；CKRM) 可實現有如大型主機的伺服器邏輯分割 (partitioning) 功能，以增進資源管理和應用彈性。

Novell SUSE LINUX Enterprise Server 9 管理手冊

- 支援最新硬體設備、進階的網路標準、先進儲存功能與連線功能，並在週邊設備和網路環境中實現防錯功能。
- 可以在不中斷系統執行的情況下利用「熱插拔」功能更換硬體，例如硬碟、處理器或其它週邊設備...等。
- 內建自動故障切換的叢集（cluster）技術。
- 內建InfiniBand高性能運算技術（High Performance Computing；HPC）。
- 內建的常用服務包括DNS、DHCP、FTP、Firewall、SSH／VPN、Proxy、Samba、NFS、LDAP、CUPS、IMAP、NTP、SLP、Postfix、PXE、SNMP、SMTP和其它網路服務與協定，因此可提供強大的基礎網路服務。
- 軟體發展能力包括應用程式和資料庫服務，例如 Apache、JBoss、Tomcat、MySQL 和 PostgreSQL。
- 支援許多第三方獨立軟體廠商（SAP、mySAP、Oracle、IBM Software Group、BEA、Computer Associates、Sun、BMC Software、Polyserve、Software AG、Veritas、SGI、Tobit、ImmUnix、Sybase...等數百家）所提供的企業級解決方案。
- 提供oracrun套件，讓企業自動化完成Oracle資料庫的環境最佳化與效能調校。
- 免費提供完整豐富的開發工具、環境、APIs與文件，讓開發者以最簡單的方法、最短的時間在 Linux 上發展應用系統；例如Eclipse、KDevelop、Anjuta、Eric and Quanta、Mono 也已經在 SUSE LINUX SDK 9中提供。
- SUSE AutoBuild的設計。

重點整理

◎ 「Unix」的意義

Unix是由貝爾實驗室(Bell Labs)的Ken Thompson與Dennis Ritchie所發展出來的；是一個多人(Multi-User)多工(Multi-Tasking)的作業系統。由於兩人先前曾參與MULTICS計劃，Unix這名字的由來就是對MULTICS開的玩笑。Unix是第一個使用高階程式語言編寫成的作業系統，利用C語言開發的原始碼具有簡單易理解的特性，加上初期AT&T對其原始碼的使用抱著開放的態度，不少大學將其作為作業系統課程的範例程式；而其高度的可移植性，衍生不同硬體平台、不同廠商的Unix。所以Unix可說是一系列作業系統的代表。

◎ 自由軟體(Free Software)的四大自由

自由軟體(Free Software)是在1980年由Richard M. Stallman所提出，他主張軟體應有四大自由：

- 任意使用的自由。
- 研究及修改以符合自己用途需求的自由。
- 拷貝給他人的自由。
- 散佈修改後軟體的自由。

◎ 「Linux」的由來及演化過程

Linux原是Linus Torvalds在1991年所開發的作業系統核心，它雖然像是Unix的翻版，但並未用Unix的原始碼，所以不受當年AT&T的版權所限。Linus Torvalds在網路上公開其原始碼，並廣邀高手加入Linux的發展工作，因此研究成果很快就散佈到世界各角落去；時至今日，kernel的版本已經出到2.6.X。Linux的設計不像傳統封閉的開發方式，而是透過網路分散式專長與迅速傳播的特性，將世界各地的高手集結在一起，共同發展成熟穩定的作業系統核心。

Novell SUSE LINUX Enterprise Server 9 管理手冊

◎ Linux、SuSE及Novell之間的關係

SuSE原為一間德國的Linux公司，在1993年由Hubert Mantel、Burchard Steinbild、Roland Dyroff及Thomas Fehr四人創立。SuSE這名字是德文「Software und System Entwicklung」的簡寫，意思為「軟體及系統開發 (Software and System Development)」。他們將Linus Torvalds的kernel加上核心工具與相關的軟體，並加入公司團隊中極具創意的系統管理模組與工具（例如：Yast），釋出一套可以完整安裝的作業系統。由於SuSE具備嚴謹、穩定、管理容易...等特性，一發展就成為德國最暢銷的Linux Distribution。

2003年，Novell收購了SuSE Linux並開始對公司進行大膽的Linux改造。2004年8月，包含完整kernel 2.6功能的SUSE LINUX Enterprise Server 9 (SLES 9)也率先推出！展現Novell以Linux作業系統為基礎，發展全系列產品線，進而成為Linux企業應用的領導者的決心與企圖心。

李明有感

做任何事情，要跨出第一步是最難的。筆者非常喜歡Nike的廣告詞「Just Do it」！希望看到這句話時，讀者已成功的踩出第一步！最後，筆者將一句相當不錯的話，送給各位讀者互勉。

If you don't wanna do it, you find an EXCUSE. If you do, you'll find a WAY.

『你若不想做，會找到一個藉口；你若想做，就會找到一個方法！』