

第一章 安裝 SuSE Linux Enterprise Server 9

章節內容：

- 1-1 Unix 的歷史
 - 1-2 Linux 的由來
 - 1-3 百家爭鳴的 Linux Distribution
 - 1-4 Novell SuSE
 - 1-5 企業級 SLES 9 的優勢
 - 1-6 安裝 SLES 9
- 重點複習
課後練習

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。

註：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。

註：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。

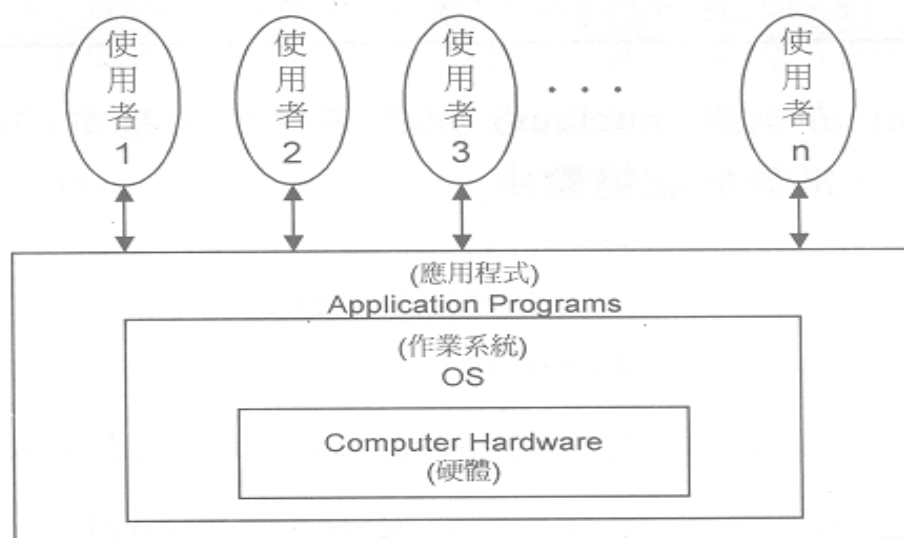


圖 1-1、作業系統與電腦硬體、應用程式關係圖

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。

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）的有力人士即決定退出 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 8 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。這台機器最後就被稱為「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】
- 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】

受限於機器的效能，「PDP-7」只能支援兩個使用者，實在稱不上多使用者 (multiuser)；當時和 Ken Thompson 一起共事的 Brian W. Kernighan 便玩弄「MUITICS」這個字，開玩笑地戲稱他們的系統其實是「UNiplexed Information and Computing System」，並縮寫為「UNICS」，後來取其諧音，就稱其為「Unix」了！提起 Unix 的名字，很多人都疑惑正確的寫法是全部大寫的「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。

註：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。

註：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；而另人驚訝的是，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。

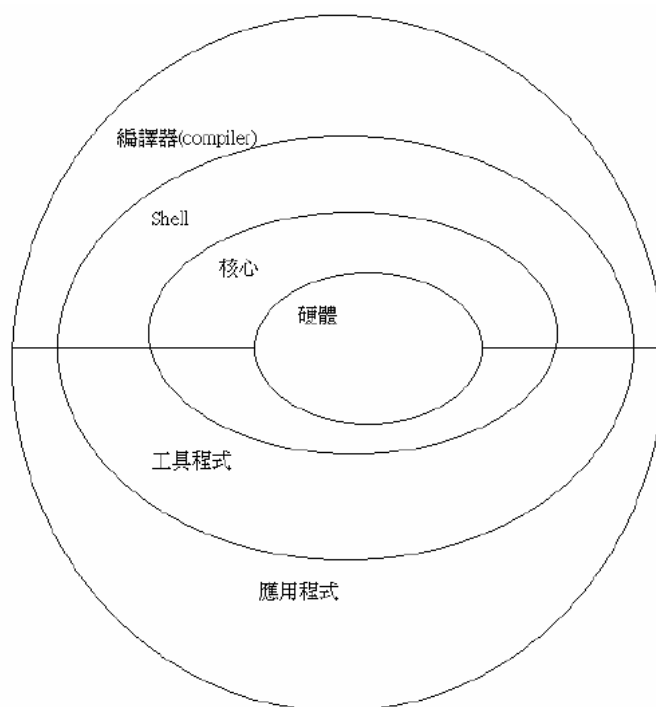


圖 1-2：作業系統簡圖

■ 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。

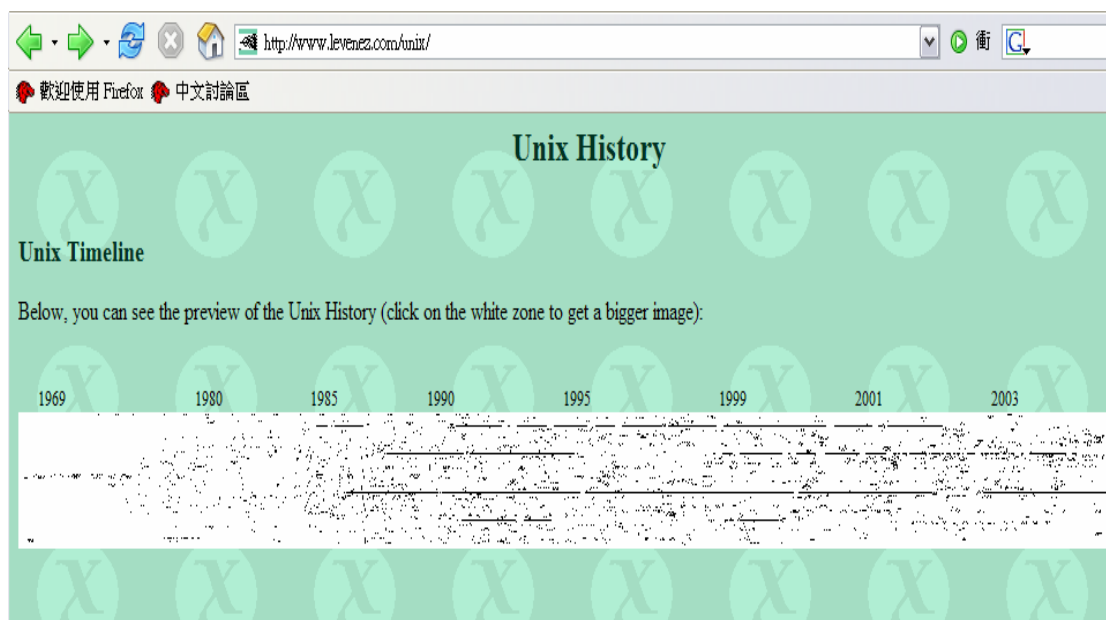


圖 1-3：介紹 Unix 歷史的網站 (<http://www.levenez.com/unix/>)

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 386/ix	2001 Security-Enhanced Linux
1992 386 BSD	1978 Interactive IS	2004 Silver OS
1986 A/UX	1988 IRIX	1983 Sinix
1989 Acom RISC iX	1991 Linux	1995 Sinix ReliantUnix
1988 Acom RISC Unix	1994 Lites	1990 Solaris 1
1990 AIX	1977 LSX	1992 Solaris 2
2000 AIX 5L	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 Atari Unix	1998 Monterey	1984 Ultrix 32M
1989 BOS	1988 more/BSD	1982 Ultrix-11
1979 BRL Unix	1983 mt Xinu	1984 Unicos
1988 BSD Net/1	1993 MVS/ESA OpenEdition	1996 Unicos/mk
1991 BSD Net/2	1993 NetBSD	1993 Unicox-max
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 Dynix	1977 PWB	1993 UnixWare
1993 Dynix/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

註：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。

- ◆ 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、具有記憶體映對檔案的虛擬記憶體子系統 (virtual memory subsystem with memory-mapped files)。
 - ksh。
 - ANSI C。
 - 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 居功闕偉！

- ◆ 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，並強調這套系統算然有一點像 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型式上就必須和 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。

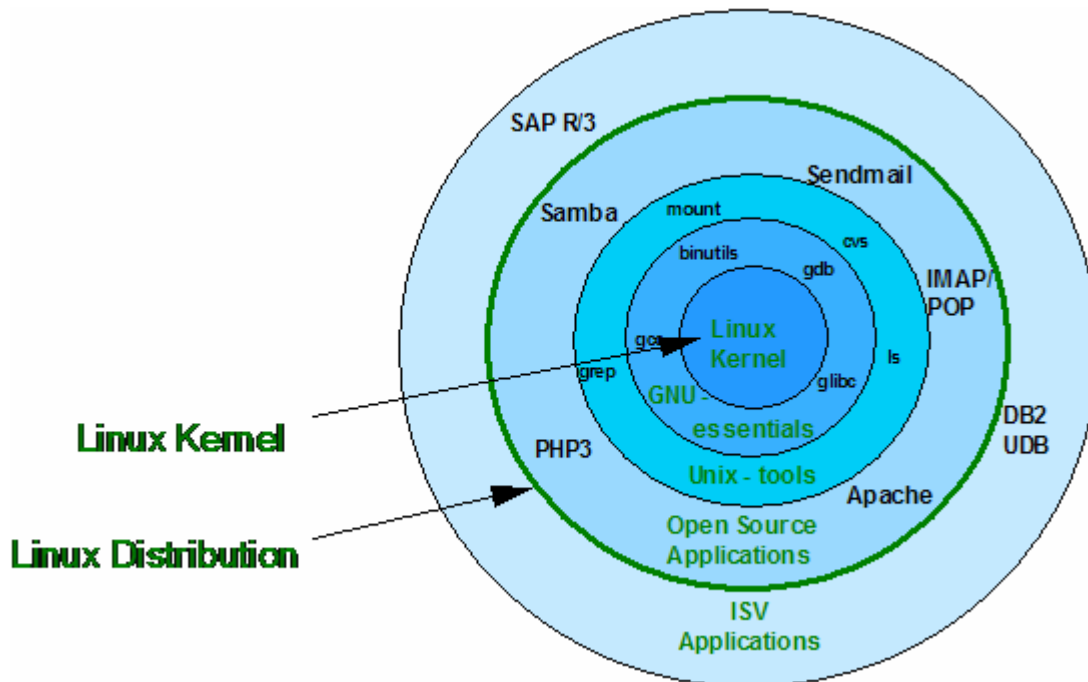


圖 1-6：Linux kernel 和 Distribution 關係圖

發展 Linux 的公司實在太多了，例如有名的 Red Hat、SuSE、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 Dyroff 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 XCom-Server；XCom-Server 與 XFree86 相容，並支援一些 XFree86 所不支援的顯示卡；而且每當 XCom 的 X Server 成熟後，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 SuSE 合併。Novell 是一家營收在 10 億美金以上的國際企業，在網路產品有著輝煌的歷史；而 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 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 台灣區總經理陳學智認為，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。

台灣網威總經理陳學智先生表示：「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術支援，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，根據報告，針對各網站對不同版本 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

■ 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。

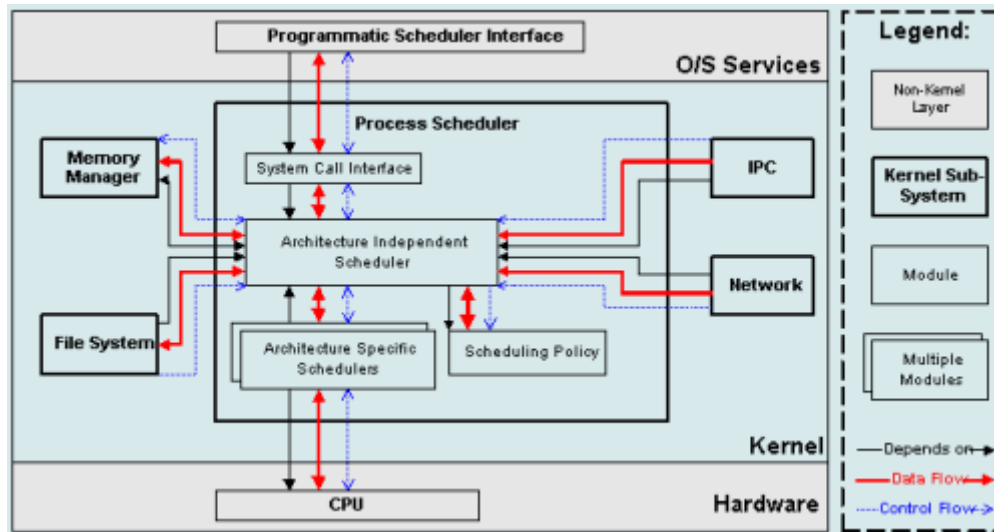


圖 1-8：scheduler 架構圖

● 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致的 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 CPU，而電腦模擬多個 CPU，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

● XFS

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 xfs 檔案系統。雖然 Linux 早已嘗試將 xfs 加入 kernel 中，但穩定度仍不如 AIX 上的 xfs；不過到 kernel 2.6 時，Linux 上的 xfs 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」可以使用 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，看來「日蝕現象」將愈來愈明顯。

◆ 電源管理

雖然在 kernel 2.4 時，ACPI(Advanced Configuration and Power Interface) 的功能一直有些問題，雖說各大 Distribution 將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。

表 1：kernel 2.4 與 kernel 2.6 比較表

Kernels	2.4.0	2.4.22	2.6.0-test7
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

■ 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) 來提供廣泛的硬體平台的執行碼。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。

■ 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。

■ 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為關鍵性軟體的安全衡量標準。

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 CPU、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。
- ◆ 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 高效能運算技術（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 orarun 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。

1-6 安裝 SLES 9

1-6-1 認識硬碟架構、MBR 及分割區

在認識 SLES 9 豐富的優勢後，各位讀者應該迫不及待地想安裝 SLES 9 來測試其強大的功能吧！先等一下！在安裝之前，我們最好對硬碟的儲存方式與分割狀況有所瞭解，對於日後管理主機才會更輕易地上手。

■ 硬碟架構

基本上，硬碟是最小的儲存單位為磁區（Sector），而數個磁區組成一個磁柱（Cylinder），最後構成整個硬碟的容量大小。一般而言，在硬碟的外殼上，通常會標示著如表 1-2 的資訊。這些數據代表什麼意義呢？我們又如何從這些數字得知此顆硬碟的大小呢？

表 1-2：硬碟容量規格資訊表

Cylinders	Heads	Sectors
16,383	16	63

硬碟是由多片金屬圓盤磁片組合而成，每片磁片的上下二面均可存放資料，每面磁片需要一個讀寫頭（head）來負責存取資料，並從「0」開始由上而下累計編號；以表 1-2 為例，上面的數字告訴我們這顆硬碟有 16 個讀寫頭（heads），也就是有 8 片磁片。

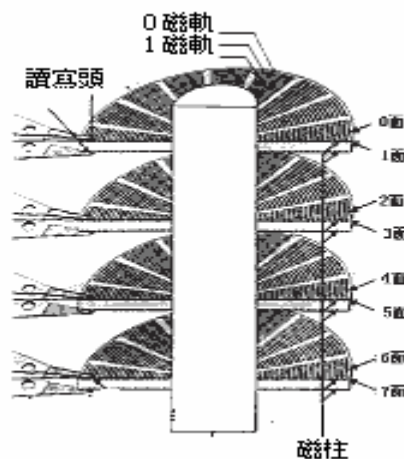


圖 1-9：硬碟內部架構

每個磁片的記錄面是由不同半徑的同心圓所組成的，我們稱這些同心圓為磁軌（Track），並將每一圈由內向外從「0」開始依序編號。每個磁軌又會切成等分的磁區（Sector），通常每個磁區可以存放 512 bytes 的資料。由表 1-2 的數字，我們可以得知此顆硬碟每個磁軌被等分成 63 個 Sectors；也就是說，每個磁軌可存放 63 X 512 bytes 的資料。

那什麼是 Cylinder（磁柱）呢？由於硬碟是由很多磁片所構成，每片磁片又有很多的磁軌，這些上下相對且編號相同的磁軌，所形成之圓柱狀磁軌組合稱之為磁柱（Cylinder），而磁柱的編號就是等於磁軌的編號。所以一個磁柱所能儲

存的空間應為：磁片數 X 2（也就是磁頭數）X 每個磁軌所能儲存的空間。

因此，表 1-2 所表示的硬碟容量，我們也就不難算出應該是：

$$\begin{aligned} & \text{Cylinders (磁柱數)} \times \text{Heads (磁頭數)} \times \text{Sectors (磁區數)} \times 512 \text{ (bytes)} \\ & 16383 \times 16 \times 63 \times 512 = 8455200768 \text{ (bytes)} \approx 8 \text{ (GB)} \end{aligned}$$

所以我們可以把整部硬碟想像成一塊「圓狀的千層糕」，以數百個同心圓（磁柱，Cylinder）將它劃分成一圈圈，並將每一圈由內向外從「0」開始依序遞增編上號碼，如此便可利用這些號碼將硬碟規劃出數個區域，每一個區域我們稱之為分割區（partition）。由此可知每個分割區一定有其起始磁柱位置（start cylinder）及結束磁柱位置（end cylinder）。

■ MBR 及分割區

在硬碟最前面的 512 bytes 稱為 MBR（Master Boot Record），這裡存放整個硬碟最重要的資訊。MBR 包含了三個部份：

◆ Boot Loader

最前面的 446 bytes 存放了一段程式碼，其最主要的作用，是要選擇從那一個分割區載入作業系統。

◆ Partition Table

後面緊接的 16 X 4 bytes 為 Partition Table。讀者可能覺得很奇怪，為什麼不直接寫 64 bytes 呢？難道是建構式數學的餘毒嗎？筆者求學那個年代可沒什麼建構式數學，九九乘法表倒也還沒忘記。之所以寫成 16 X 4 的樣式，是因為每個分割區資訊需佔用掉 16 bytes（即 start cylinder、end cylinder 的位置及其他相關資訊）。MBR 利用 16 X 4 bytes 來存放 Partition Table，意謂著 Partition Table 最多只能記錄 4 個。

◆ Magic Number

最後 2 個 bytes 為 MBR 結束碼。

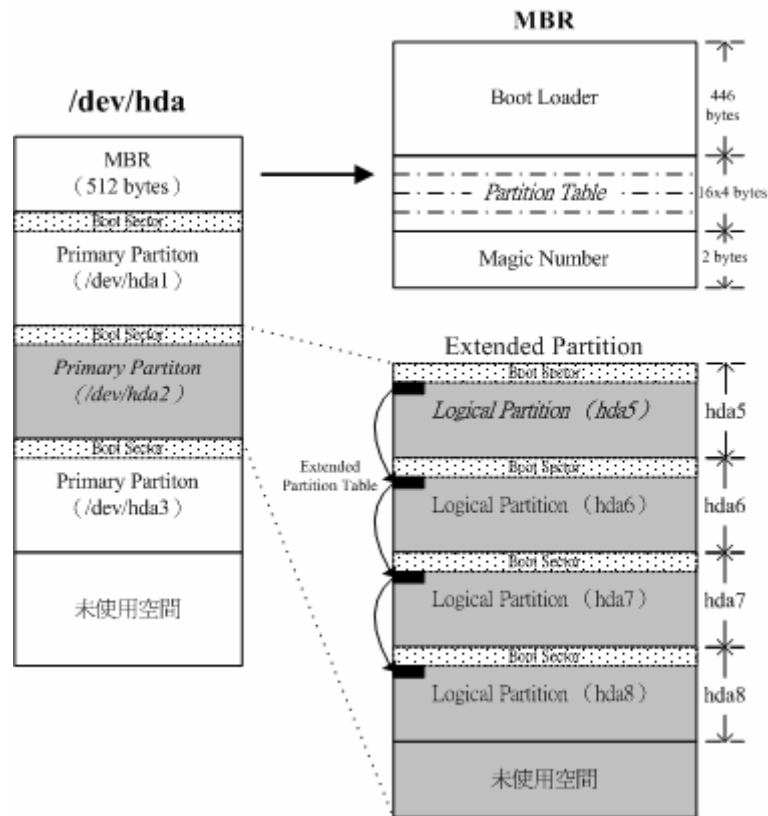


圖 1-10：MBR 與分割區概念圖

註：在 Linux 中，IDE 硬碟的代號為 hd#，SCSI 硬碟的代號為 sd#。一般而言，個人電腦有兩個 IDE 介面：

IDE1 (Primary IDE) 所接的 Master 硬碟代號為 hda

IDE1 (Primary IDE) 所接的 Slave 硬碟代號為 hdb

IDE2 (Slave IDE) 所接的 Master 硬碟代號為 hdc

IDE2 (Slave IDE) 所接的 Slave 硬碟代號為 hdd

分割區可分為三個種類：分別是「Primary 分割區」、「Extended 分割區」和「Logic 分割區」。怎麼區分這些類別呢？

◆ Primary 分割區

因為 partition table 的大小為 16 X 4 bytes，所以一顆硬碟最多只能切出 4 個分割區，這 4 個分割區就是 Primary 分割區。

◆ Extended 分割區

4 個分割區是不敷使用的！為了解決分割區切割數量的限制，所以我們取出其中一個分割區（註：只能用其中的一個分割區）來擴充，這個分割區就稱為 Extended 分割區。Extended 分割區不一定得用 hda4，以圖 1-10 為例，

即是使用 hda2。

◆ Logical 分割區

Extended 分割區是無法直接使用的，必須再細分為 Logical 分割區才可用來存放資料。Logical 分割區的起始位置及結束位置記錄在每個 Logical 分割區的第一個磁區（稱為 Extended partition table）。如果要建立一個以上的 Logical 分割區，則分割程式會自動在 Extended Partition Table 上產生一個分割區，如此不斷地延續下去，猶如一條鏈子，直到 Extended 分割區所涵蓋之區域用盡，故可將一部硬碟切成 N 塊區域使用。

1-6-2 安裝 SLES 9

■ 安裝時硬碟分割配置的考量

為什麼安裝 Linux 時得指定每個檔案系統的大小？筆者認為最主要有幾個原因：

◆ 備份與還原：

因為系統管理者可以針對分割區備份重要的資料；例如我們將「/home（在 Linux 中，此路徑用來存放一般使用者的資料）」額外對應至一個分割區，系統管理者若是強調使用者資料的重要性，就可以只針對這個分割區進行頻率較高的備份。

◆ 磁碟限量（Quota）：

磁碟限量（Quota）限制使用者或群組可使用的磁碟空間或可建立的檔案數。因為 Quota 是針對分割區限制，所以管理者會將需要限制的目錄額外對應不同的分割區。舉例來說，假設系統管理者架設 Mail Server，每個使用者的郵件信箱是存放於「/var/spool/mail」目錄內，若將此目錄對應到獨立的分割區，就可以單獨限制每個人的郵件信箱的大小。

不管使用哪種 Linux Distribution，通常硬碟分割大概都會提供「自動分割」及「手動分割」這兩種方式。筆者建議讀者選擇手動分割，相信會對整個檔案系統的配置情形更為瞭解。而剛接觸 Linux 環境的新手，通常不知該如何規劃硬碟的分割；筆者的經驗是—「各取所需」，根據實務需求來決定檔案系統的大小。

不過這話說來容易，做起來卻不簡單，新手或許對於 Linux 可以提供的服務都還不太認識，又怎麼根據需求來分割硬碟？因此，讀者若完全沒概念，建議只要切割「/」及「Swap」即可，而「/」最好有 8GB 以上的空間，「Swap」通

常 1GB 就綽綽有餘。此外，不要用畢硬碟所有的空間，預留的分割區可以用來備份其他分割區的資料，或拿來學習如何新增檔案系統。

想要體驗 SLES 9 (SuSE Linux Enterprise Server 9) 強大的功能，第一件事就是將 SLES 9 安裝成功；新手不要抱著一次安裝就應該完全符合需求的心態，隨著對 Linux 愈來愈瞭解，知道可提供的服務與自己的需求，便會對硬碟空間的分配更得心應手。Just Do it ! 不要排斥重新安裝系統，凡事不可能一蹴而成。

■ 逐步安裝 SLES 9

為了可以順利進行本書相關的實例演練，筆者將系統規劃如下：

◆ **目的：**
安裝練習用 SLES 9 Server 。

◆ **套件：**
使用預設套件。

◆ **硬碟分割：**

分割區	目錄	空間
/dev/hda1	/	8GB
/dev/hda2	/home	1GB
/dev/hda3	Swap	1GB

◆ **網路設定：**

主機名稱	testing.suse.com
IP 位址	192.168.123.10
子網路遮罩	255.255.255.0
預設開道	192.168.123.2
DNS 伺服器	168.95.1.1

【Preparation】

步驟 1、設定開機次序並從光碟開機。

放入 SLES 9 第一片光碟後，請調整 BIOS 以設定開機的次序，選擇從光碟機開機。接著會看到如圖 1-11 有著 Novell SuSE 綠色變色龍 geeko 的開機畫面。



圖 1-11：SLES 9 的開機畫面

步驟 2、選擇安裝方式。

靜待歡迎畫面結束後，會出現如圖 1-12 的選單，請選擇第二項「Installation」。若讀者欲利用文字介面進行安裝，可按下「F2」，便可出現選單變更為文字安裝模式或調整安裝畫面的解析度。若想利用其他安裝的方式（FTP、NFS、SMB...等），可按下「F3」更改安裝來源。

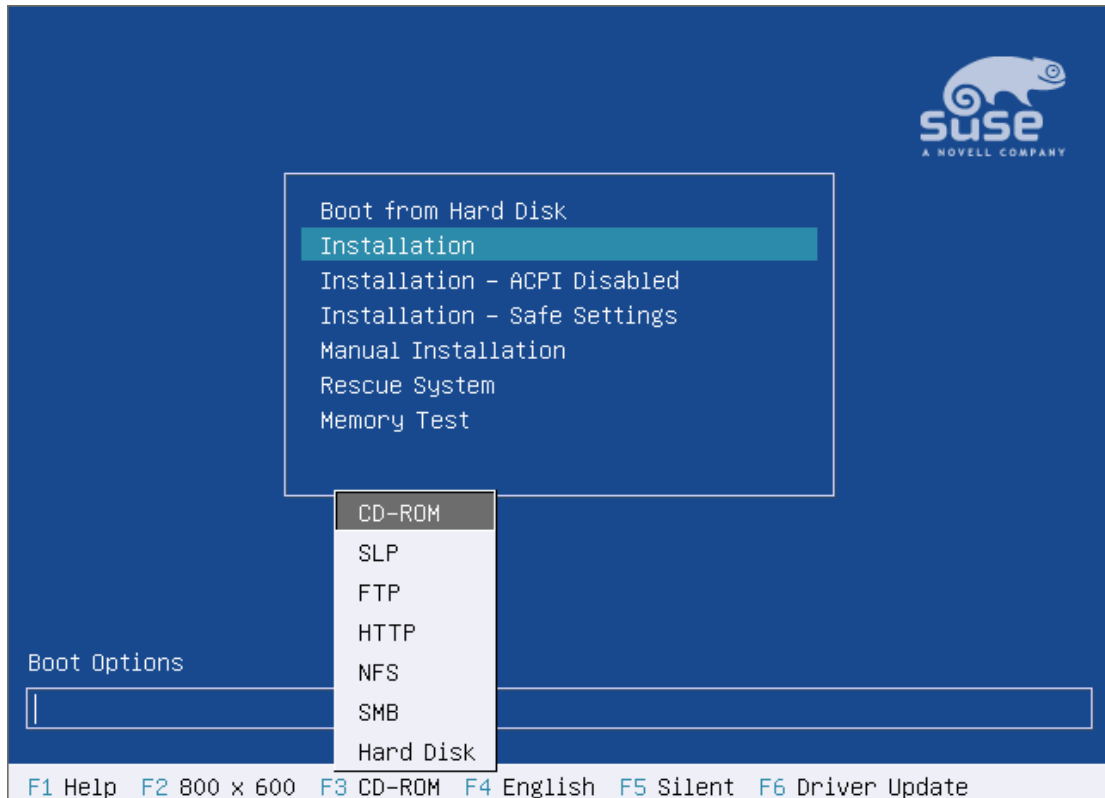


圖 1-12：SLES 9 開機選單

步驟 3、版權聲明。

接受 Novell SLES 9 的版權聲明，才可以繼續安裝。

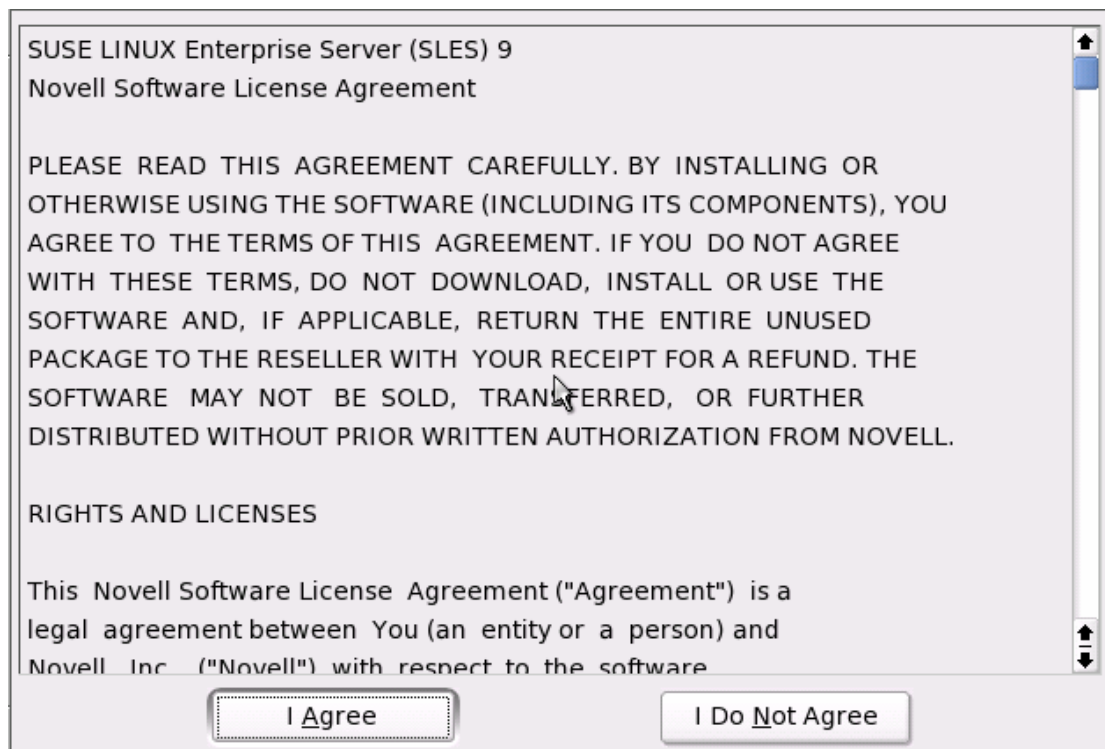


圖 1-13：SLES 9 版權聲明畫面

【Base Installation】

步驟 4、選擇語系。

大家都是愛台灣的資訊人，當然選「繁體中文」囉！選好後請點選「Accept」。

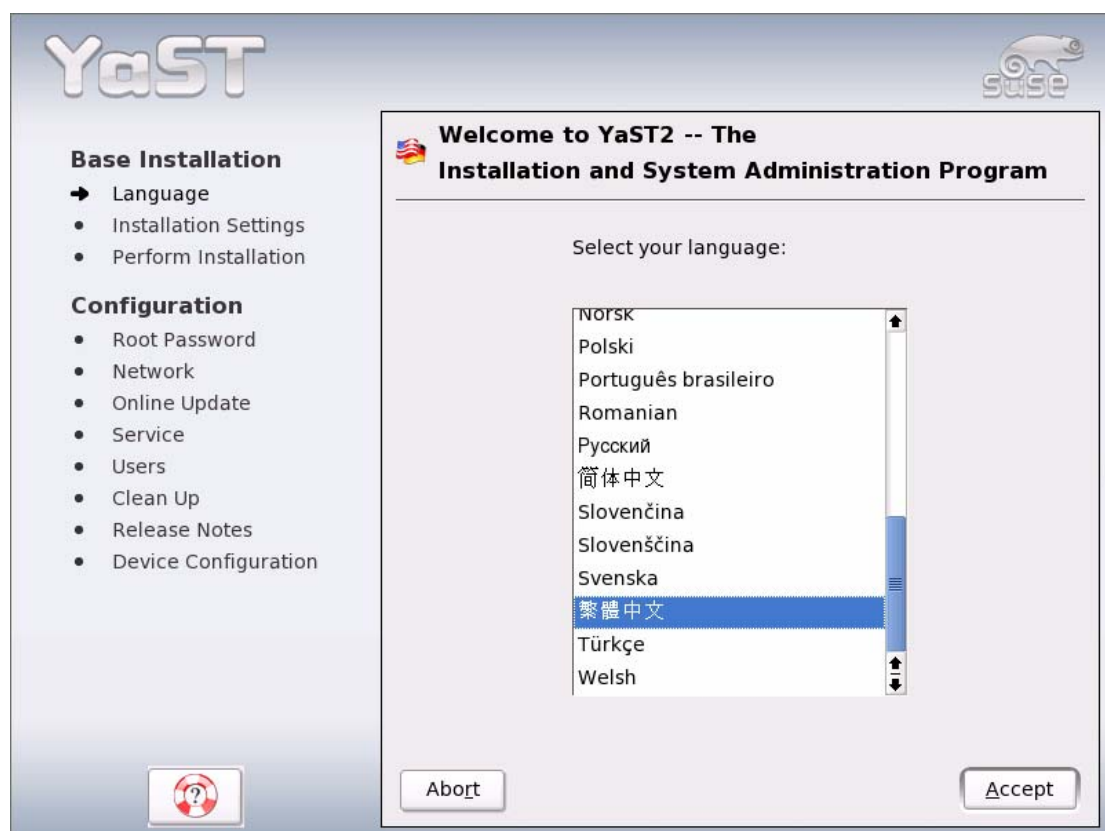


圖 1-14：選擇語系畫面

步驟 5、安裝設定。

SLES 9 的安裝程式會自動偵測硬體資訊並幫忙設定，若想調整，可點選下方的「Change...」選單更改相關設定。例如：欲自訂硬碟分割區，請點選「Change...」選單後，選擇「Partitioning...」。

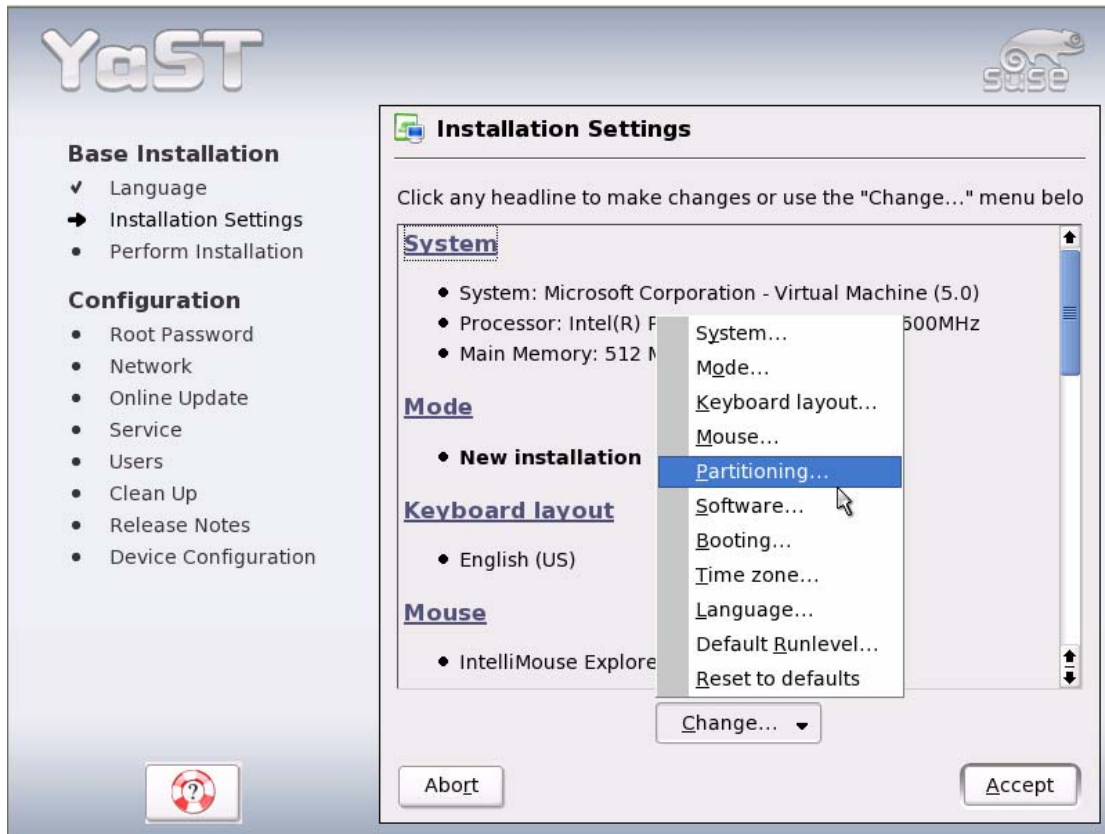


圖 1-15：安裝設定畫面

步驟 6、設定檔案系統及硬碟分割區

檔案系統及硬碟分割區的設定是安裝時最重要的步驟；SLES 9 提供三種方法分割硬碟，如圖 1-16：

◆ 「Accept proposal as-is」：

採用 SLES 9 預設值來分割。SLES 9 建議分割的方式顯示於畫面上方，若採用此選項，Swap 分割區會安裝於「/dev/hda1」，並配置 1008.4MB；root 分割區會安裝於「/dev/hda2」，使用 reiser 檔案系統格式，並配置 15GB 的空間。

◆ 「Base partition setup on this proposal」：

根據系統的預設值略作修改。

◆ 「Create custom partition setup」：

由使用者自行訂定檔案系統及對應硬碟分割區大小。

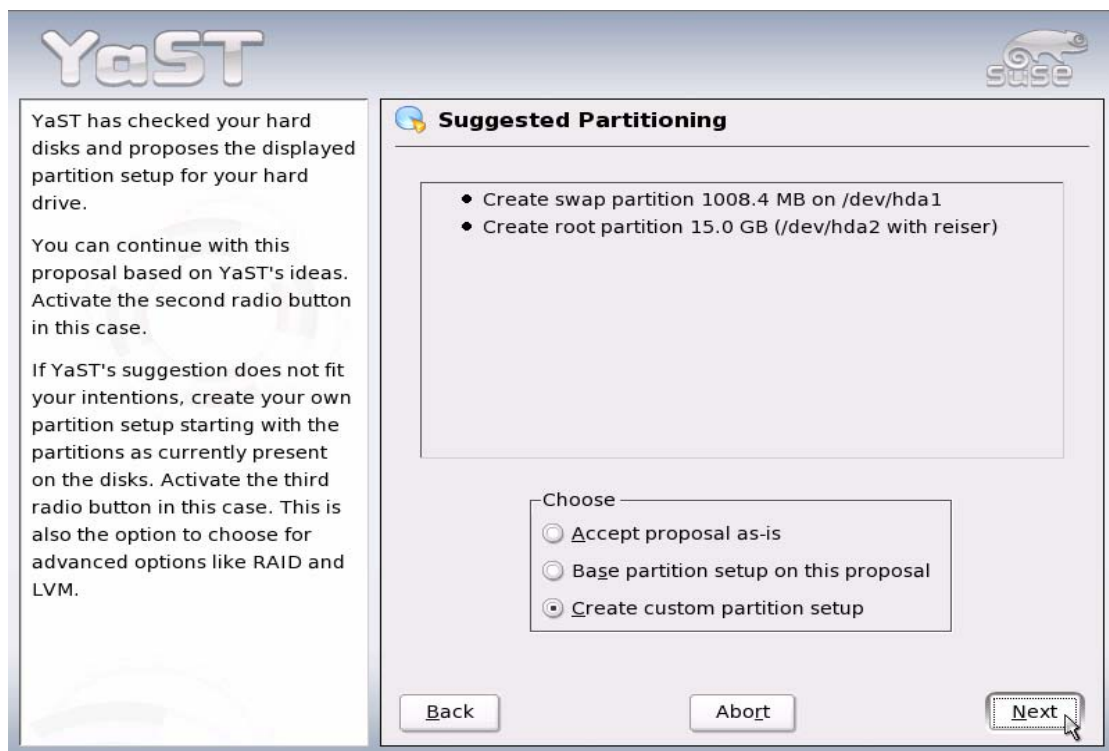


圖 1-16：分割硬碟方法選項

由於前兩個項目不符合我們規劃的硬碟分割需求，故選擇「**Create custom partition setup**」才能依先前的條件配置。點選此項目後按「Next」，將會出現圖 1-17 的畫面，請選擇「**Custom partitioning – for experts**」以完整控制。

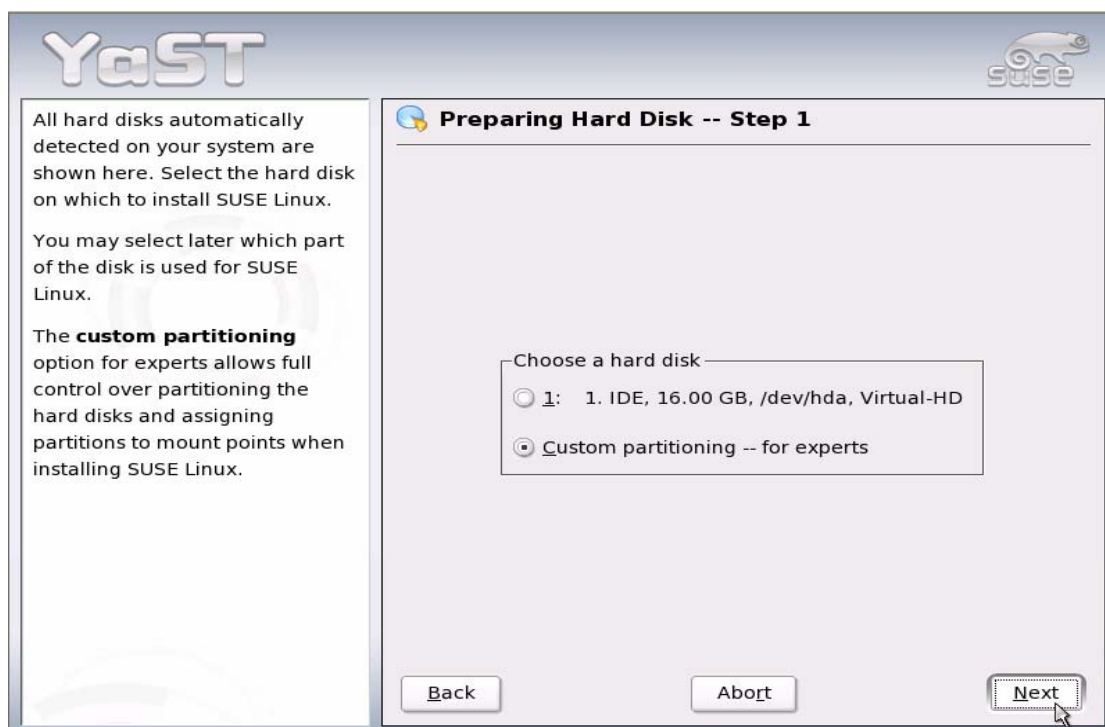


圖 1-17：自訂分割區畫面

接著會顯示如圖 1-18 的「Expert Partitioner」畫面，在此頁面即可設定各分割區的大小、檔案格式、掛載點...等。

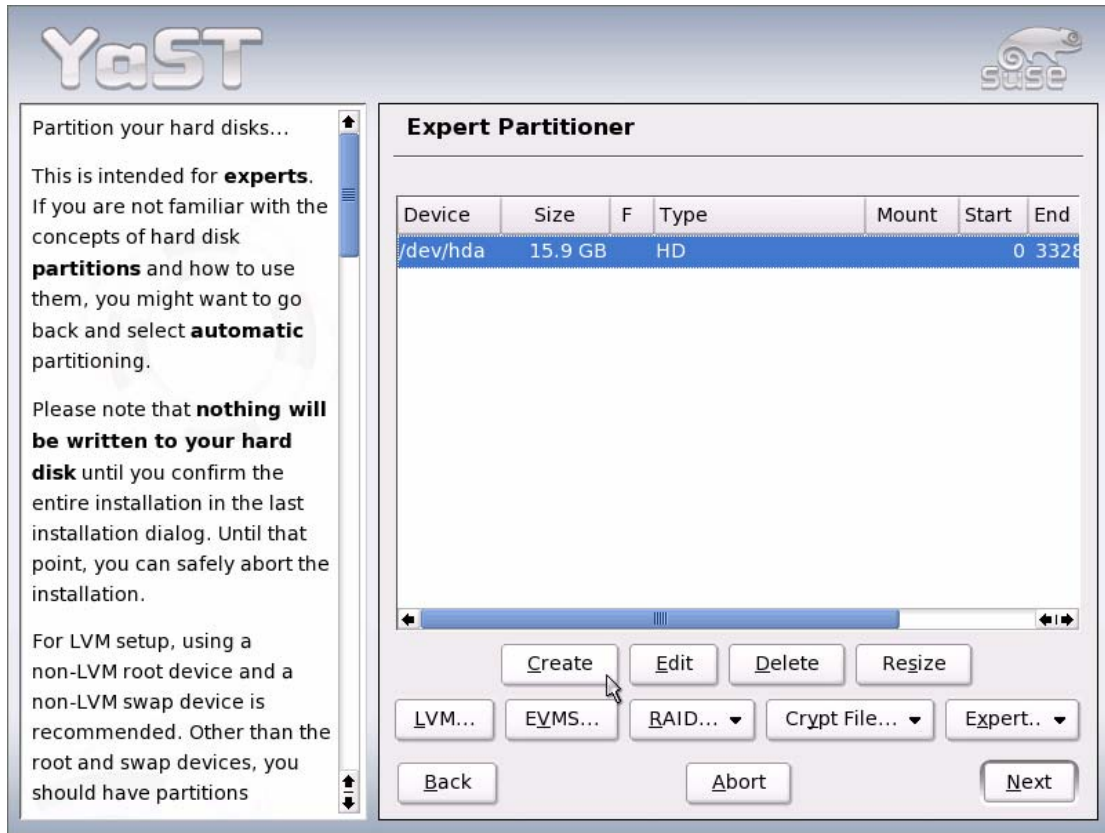


圖 1-18 : Expert Partitioner 畫面

依照先前筆者規劃的檔案系統及對應硬碟分割區大小，要設定這 3 個分割區的方式如下：

分割區	目錄	空間
/dev/hda1	/	8GB
/dev/hda2	/home	1GB
/dev/hda3	Swap	1GB

◆ 建立「 / 」檔案系統

要建立新的分割區，請點選按下「Create」，便會出現如圖 1-19 選擇「Primary Partition」或「Extended Partition」畫面。

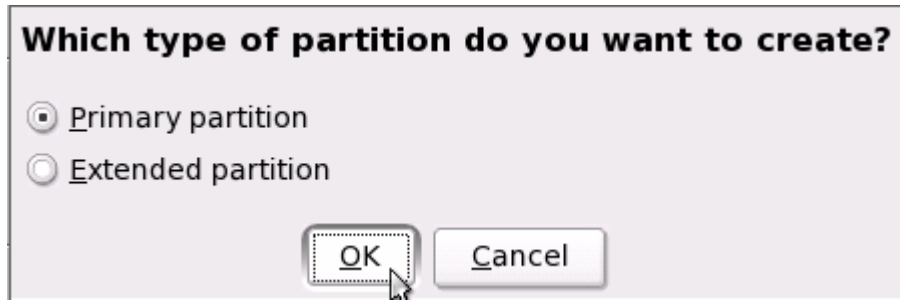


圖 1-19：選擇「Primary Partition」或「Extended Partition」

Primary 分割區最多有 4 個，由於我們只需要 3 個分割區，Primary 分割區尚且足夠，所以選擇「Primary partition」。接著出現如圖 1-20 的畫面，用以指定檔案系統類型、檔案大小、掛載點...等資訊。Size 區域主要是要調整分割區大小；Start cylinder 請採用預設值，End cylinder 填入「+8G」，即可指定「/」檔案系統大小為 8GB。Mount Point 請設定為「/」，並使用「Reiser」的檔案系統類型格式。

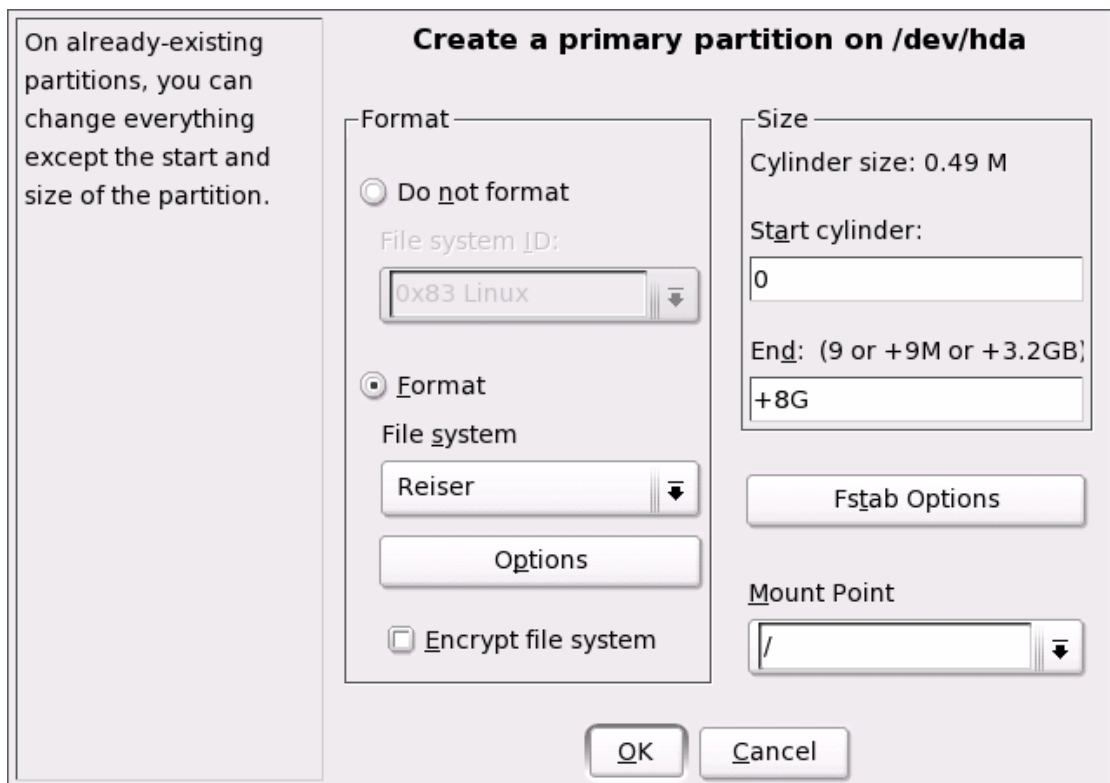


圖 1-20：建立「/」檔案系統畫面

◆ 建立「/home」檔案系統

如同建立「/」檔案系統的方式，在 Expert Partitioner 畫面按下「Create」後，選擇「Primary Partition」，即可跳至如圖 1-21 的設定畫面。較為不同的，是檔案系統大小指定為 1GB；Start cylinder 同樣採用預設值，End

cylinder 填入「+1G」即可。Mount point 請利用下拉式選單選擇「/home」。

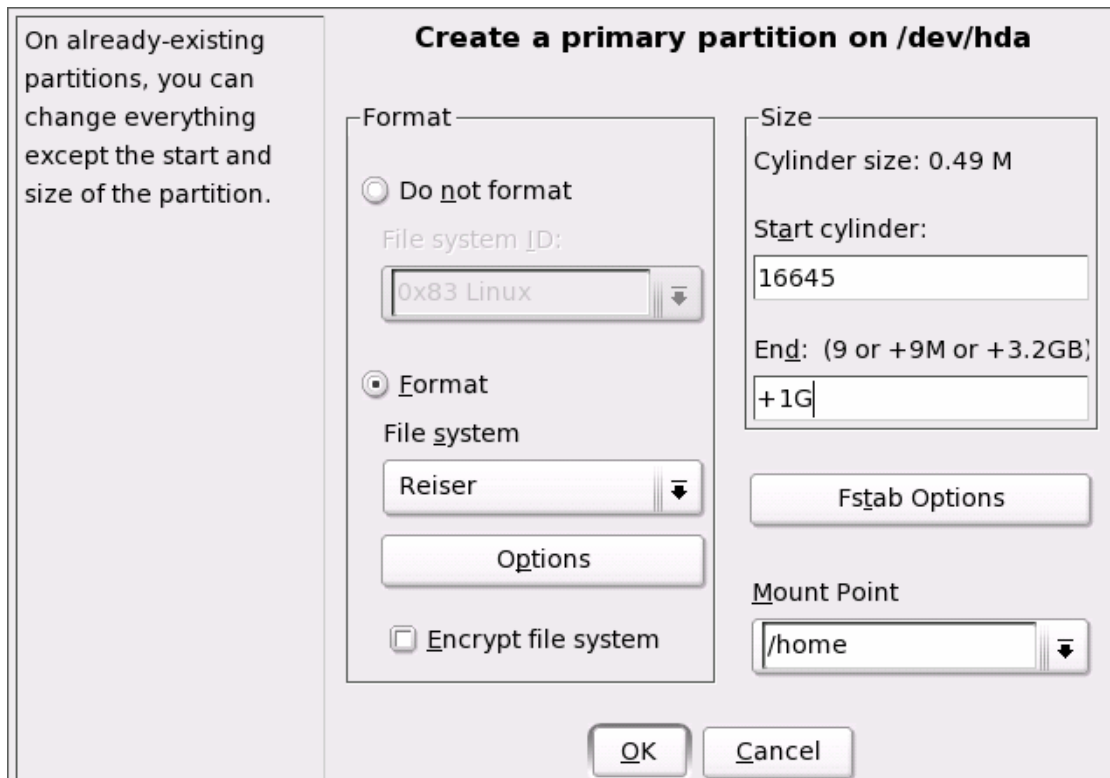


圖 1-21：建立「/home」檔案系統畫面

◆ 建立 Swap 分割區

Swap 分割區的用途是拿來作『虛擬記憶體』。什麼是『虛擬記憶體』？當系統實體記憶體不敷使用的時候，作業系統可以使用硬碟來模擬記憶體的資料存取，將實體記憶體不常被存取的資料丟到虛擬記憶體。一般書上常建議 Swap 分割區為實體記憶體的兩倍大，筆者覺得可根據需求來決定，若是硬體的記憶體已高達 1GB，而此系統的負載又不大，其實無需切割到 2GB。

同樣在 Expert Partitioner 畫面按下「Create」，選擇「Primary Partition」，後跳至如圖 1-22 的設定畫面。指定分割區大小為 1G，並利用下拉式選單將 File system 選項指定為「Swap」。

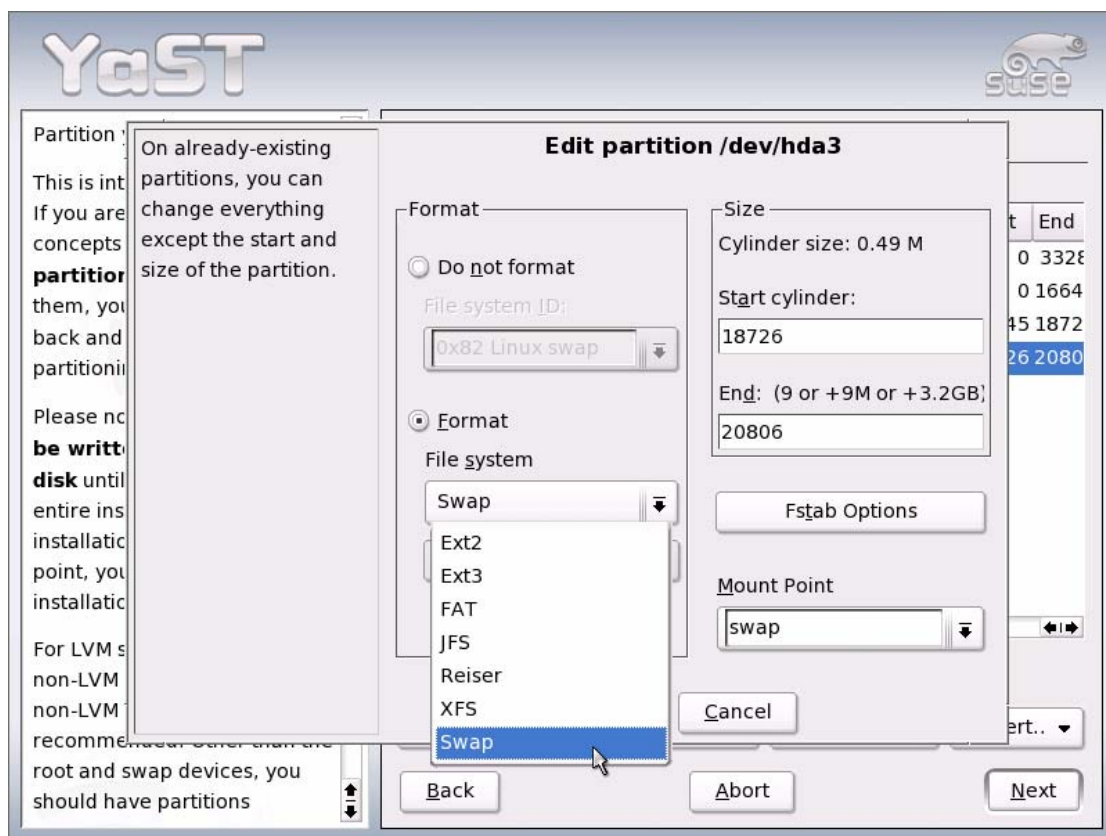


圖 1-22：建立「Swap 分割區」畫面

檔案系統及硬碟分割區全部設定完成後，可看到如圖 1-23 的畫面，然後按下「Next」。

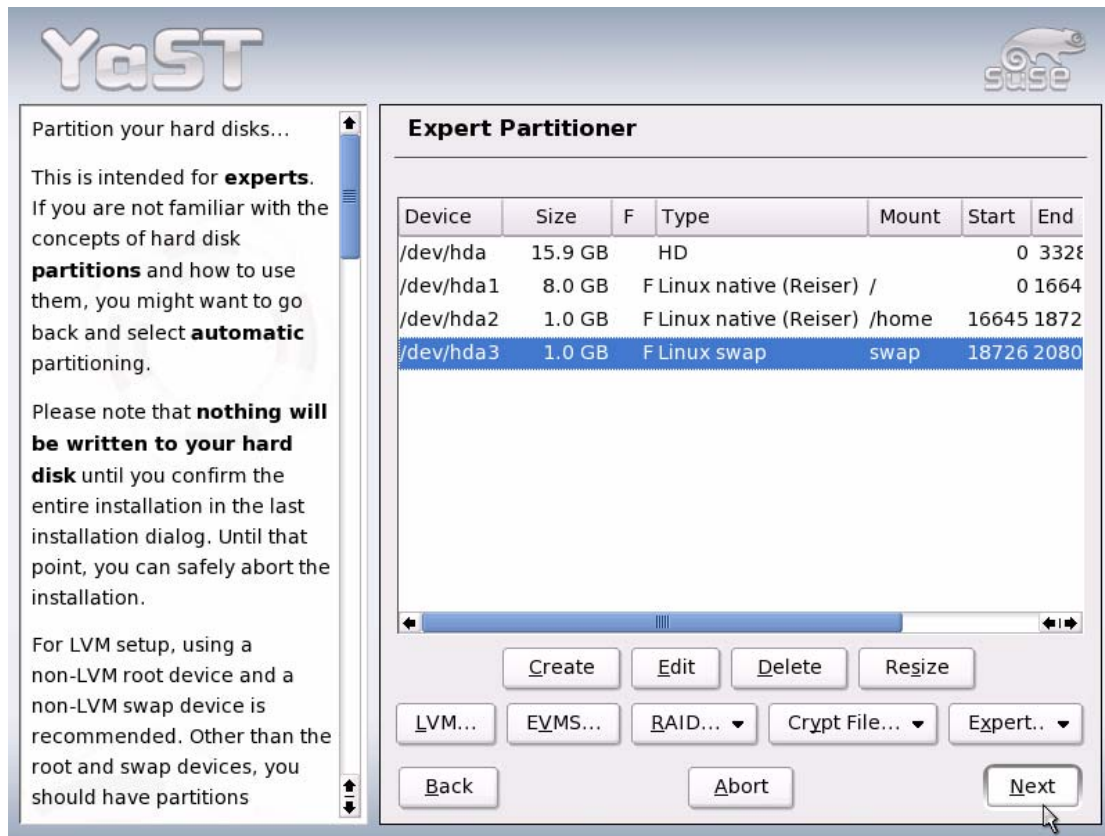


圖 1-23：檔案系統及硬碟分割區設定完成畫面

步驟 7、修改時區設定。

完成檔案系統及硬碟分割區設定後會跳回安裝設定畫面，點選「Change...」下拉選單，選取「Time zone...」可作時區的調整，如圖 1-24。

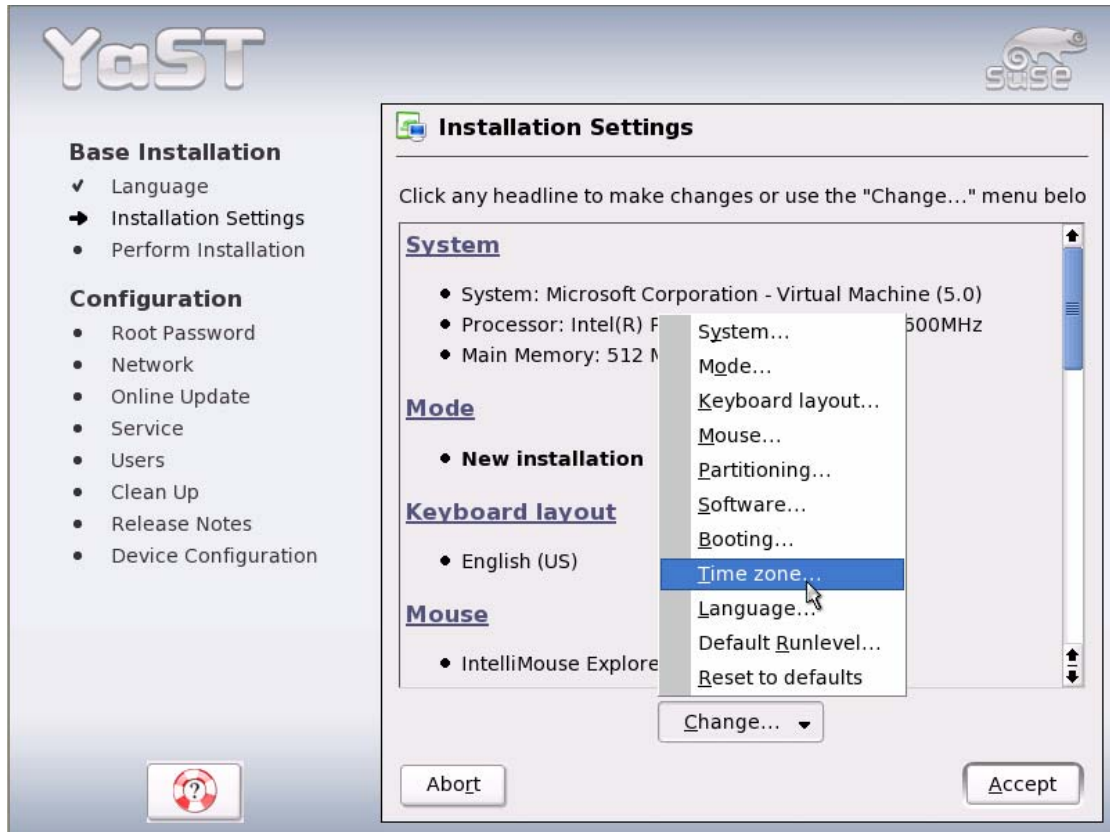


圖 1-24：修改時區

接著是如圖 1-25 的設定畫面。請在「Region」區域指定為 Asia，「Time Zone」選取 Taipei，下方「Hardware clock set to」的選項指定為 local time。請不要選擇 UTC（英格蘭格林威治標準時間），否則系統時間會被設定為格林威治標準時間，這與台灣的時間不符合。按下「Accept」鍵後又會跳回安裝設定畫面。

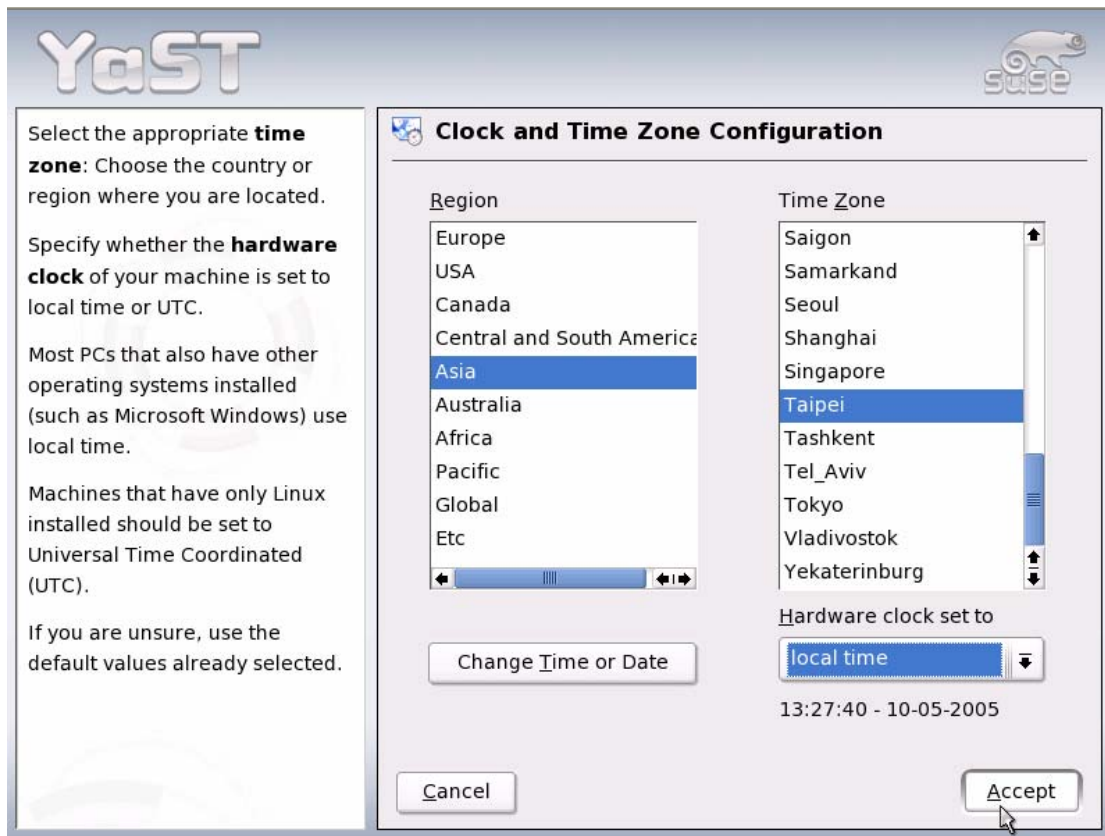


圖 1-25：指定時區

步驟 8、開始安裝。

若沒有其他欲更改的設定，按下「Accept」後便會出現確認安裝的畫面，如圖 1-26。別猶豫，點選「Yes, install」就開始安裝了（如圖 1-27）！此過程依照系統狀況需要一些時間，並會提醒更換安裝光碟片。

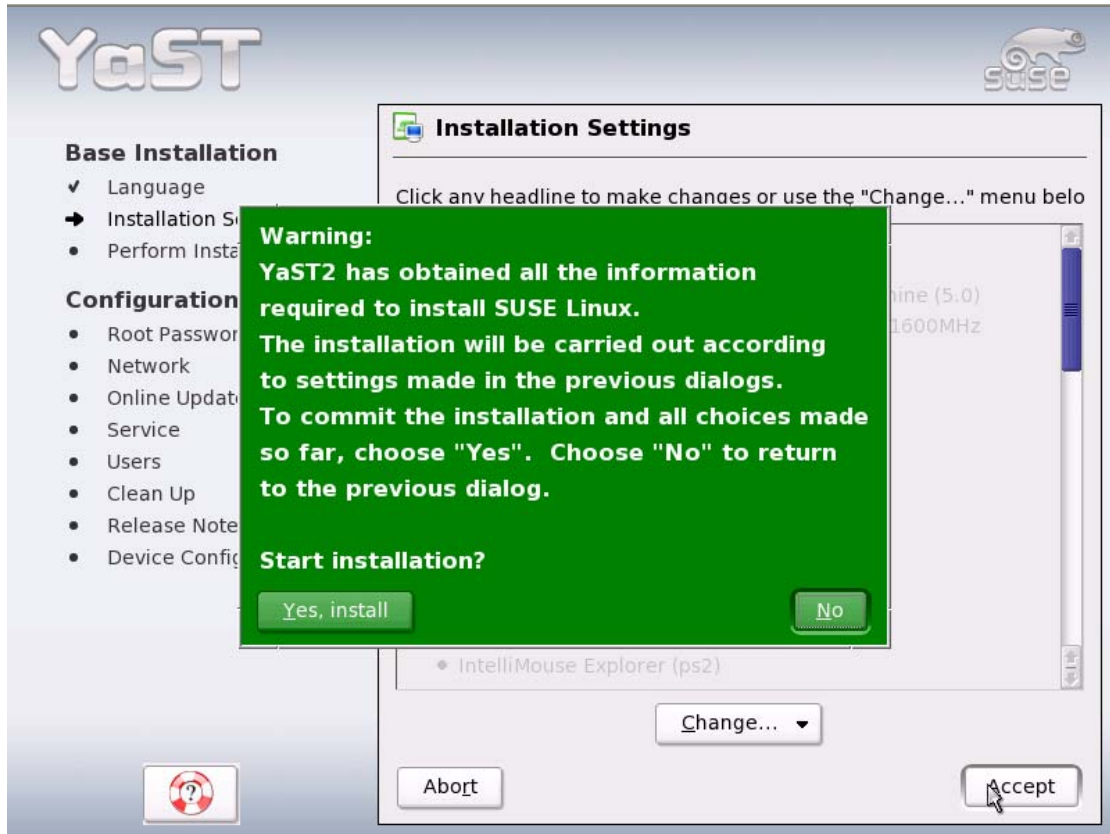


圖 1-26：安裝確認畫面

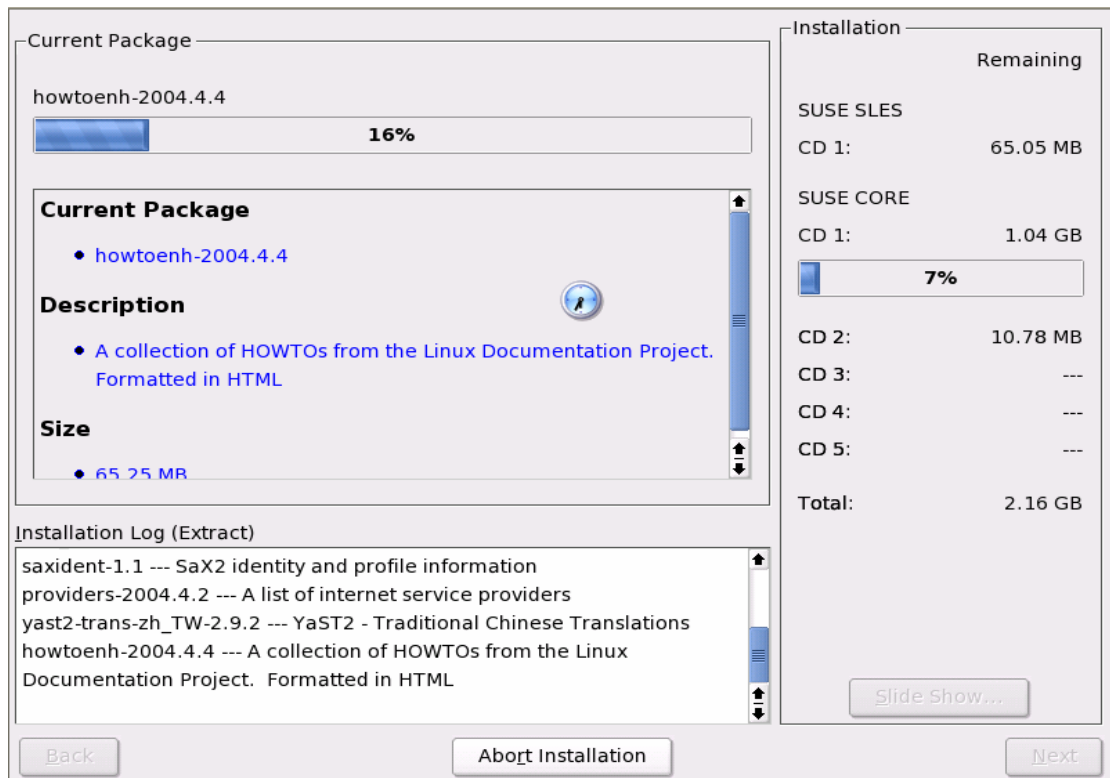


圖 1-27：安裝過程

【Configuration】

步驟 9、設定 root 密碼。

如圖 1-28，接著設定 root 的密碼。畫面上方翻譯的「根基用戶 (root) 的通行口令」是彼岸的用法，台灣用戶應該還是習慣用「密碼」代替「通行口令」吧！SLES 9 中的 root 帳號就像在 Windows 系統中的 administrator 帳號一樣，以此帳號登入將對整個系統有完全的掌控權。



圖 1-28：設定 root 的密碼

步驟 10、網路設定。

網路的設定值預設是採用 DHCP，由於我們規劃使用固定 IP，可點選「Change...」下拉選單，選擇「Network Interfaces...」調整，如圖 1-29。

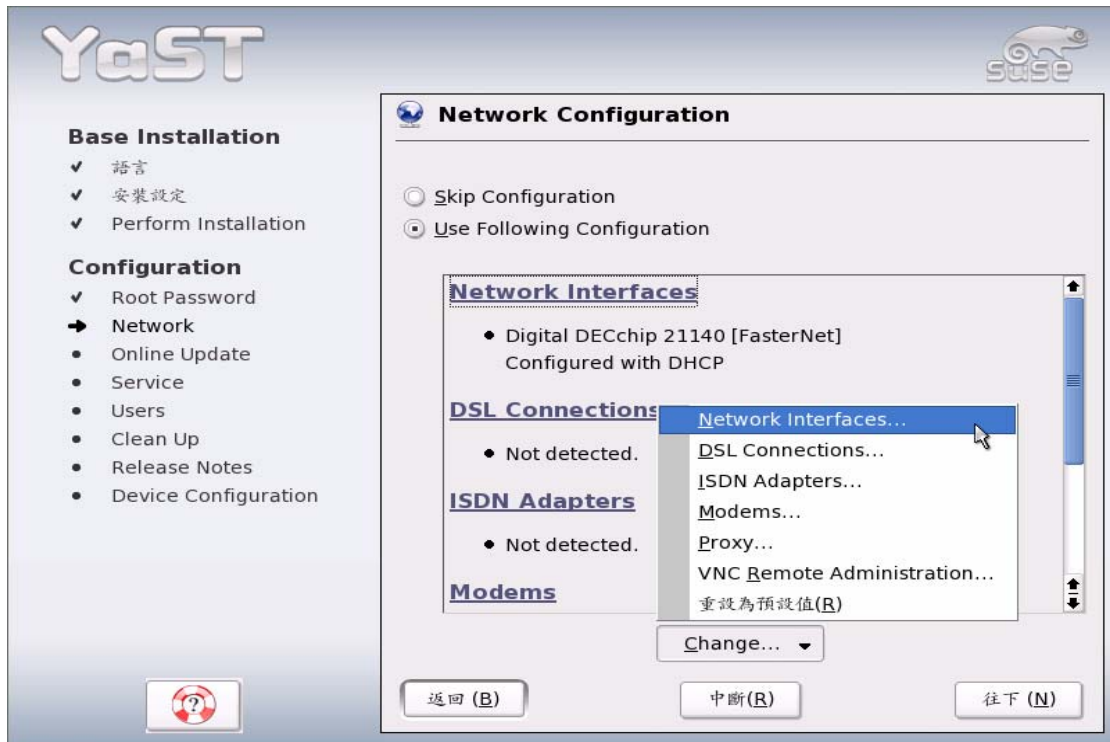


圖 1-29：更改「Network Interfaces...」設定

接著出現圖 1-30 的畫面，畫面下方顯示 Linux 有正常測偵的網路卡，點選「Change...」，可以針對此張網路卡設定 IP 位址、預設閘道器...等值。接著會跳至網路卡組態設定畫面，如圖 1-31。

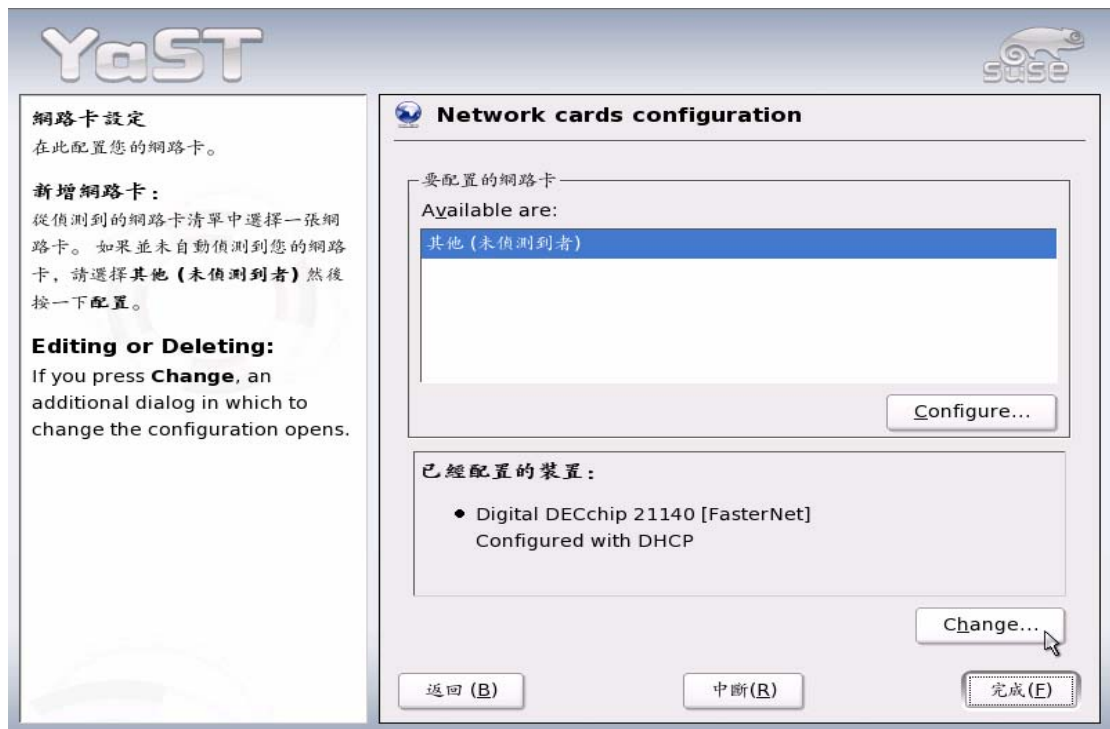


圖 1-30：更改網路卡的設定



圖 1-31：網路卡組態設定

依照先前筆者規劃條件，要設定這些網路卡組態的方式如下：

主機名稱	testing.suse.com
IP 位址	192.168.123.10
子網路遮罩	255.255.255.0
預設閘道	192.168.123.2
DNS 伺服器	168.95.1.1

◆ IP 位址及子網路遮罩

在網路卡組態設定畫面中（圖 1-31），按下「Edit」便可針對此張網卡做相關的異動。請於 IP 位址欄設定為「192.168.123.10（私有 IP）」，子網路遮罩則設定「255.255.255.0」，如圖 1-32。

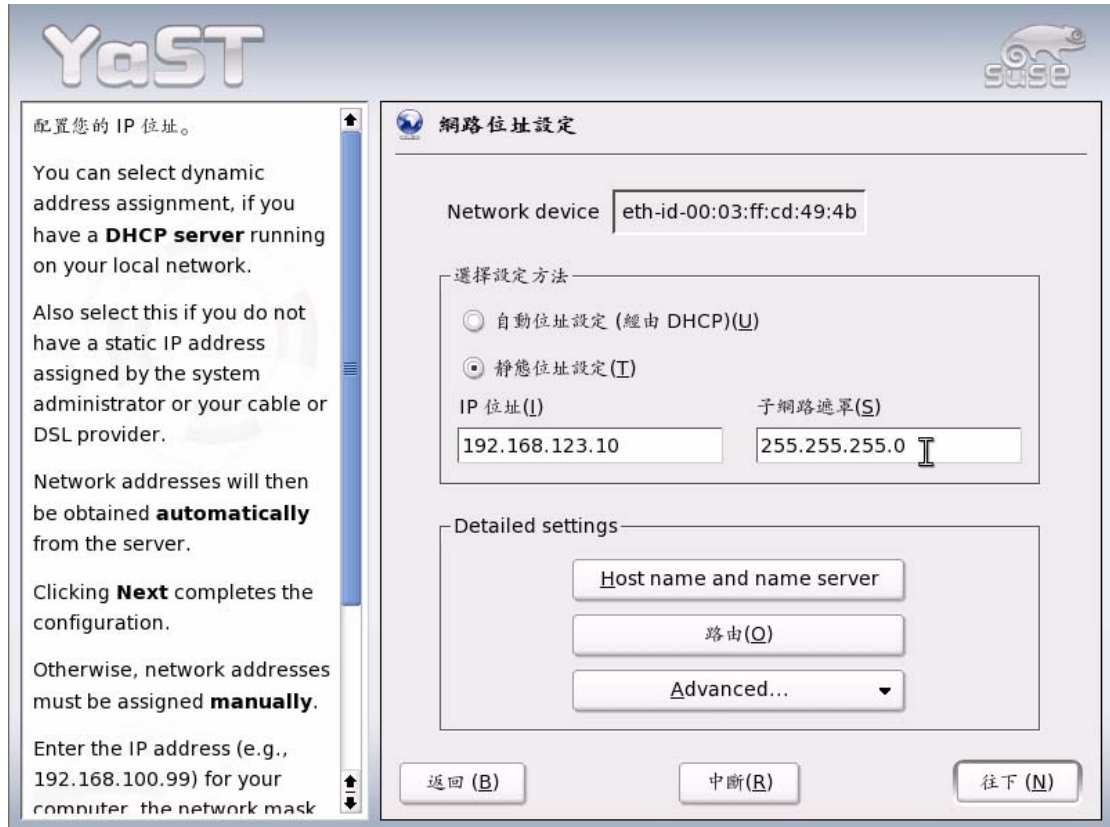


圖 1-32：網路位址設定

◆ 設定主機名稱、網域名稱、名稱伺服器及網域搜尋清單

在網路位址設定畫面中（圖 1-32）中，點選「Host name and name server」即可設定主機名稱與名稱伺服器。如圖 1-33，請在 Host Name 欄位輸入「testing」，Domain Name 欄位輸入「suse.com」。在實際的網際網路上，主機名稱應該不能重複，在此我們只是做測試，所以自行填上喜愛的 Domain 名稱。名稱伺服器所設定的「168.95.1.1」是 Hinet 的 DNS 伺服器，Domain Search 請輸入「suse.com」。設定完成後接著按下「行 (O)」便回到圖 1-32 的網路位址設定畫面。



圖 1-33：設定主機名稱、網域名稱、名稱伺服器及網域搜尋清單

◆ 設定預設閘道器

在網路位址設定畫面中（圖 1-32）中，點選「路由（O）」即可指定預設閘道器，如圖 1-34。預設通訊閘的設定要根據你個人的環境調整，如果不知道自己的網路狀況，就設定跟我一樣吧！反正安裝之後隨時可再自行修改。設定完成後，按下「行（O）」進行下一個步驟。

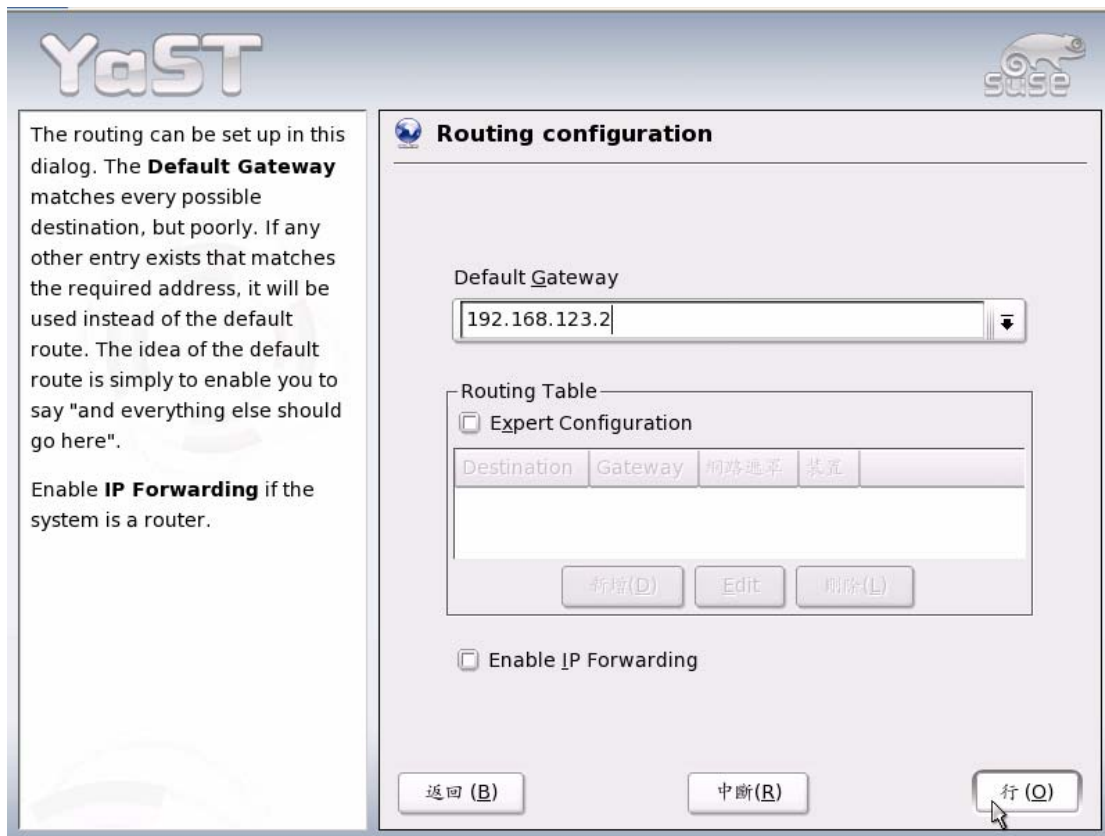


圖 1-34：預設開道器設定畫面

◆ 開放 VNC 遠端管理

雖說其他的 Linux 發行商也有支援 VNC 遠端管理，不過像 SLES 9 如此方便及美觀的 VNC 遠端管理可是少見。在不考慮安全性下，筆者建議開放 VNC 遠端管理，待安裝完畢後，我們再來體會它的好處。至於如何開放？如圖 1-35，回到網路設定畫面後，點選畫面下方的「Change...」，選擇「VNC Remote Administration...」後，即會出現如圖 1-36 的 Remote Administration 畫面，點選「Allow Remote Administration」即可。

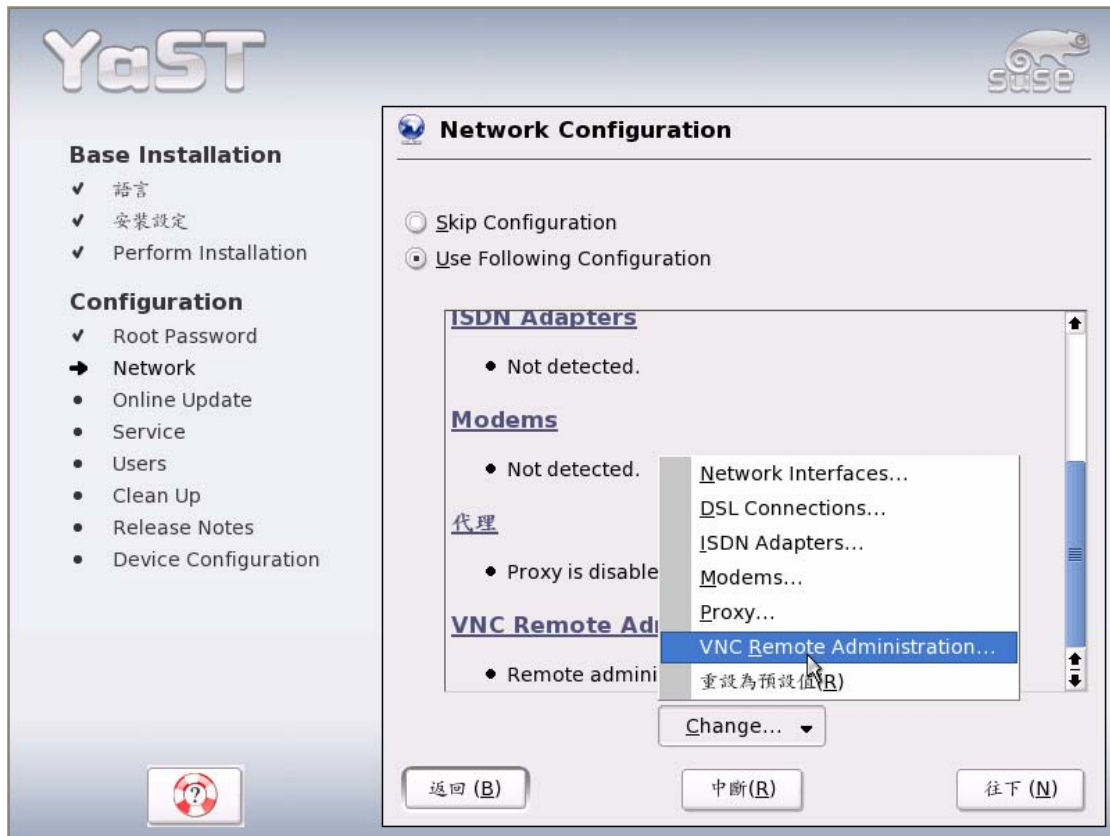


圖 1-35：更改「VNC Remote Administration...」設定



圖 1-36：開放 VNC 遠端管理

◆ 測試網路連線

網路相關設定完成後，會出現網路連線測試畫面，如圖 1-37。若是讀者覺得有需要進行測試，請勾選「Yes, Test Connection to the Internet」。

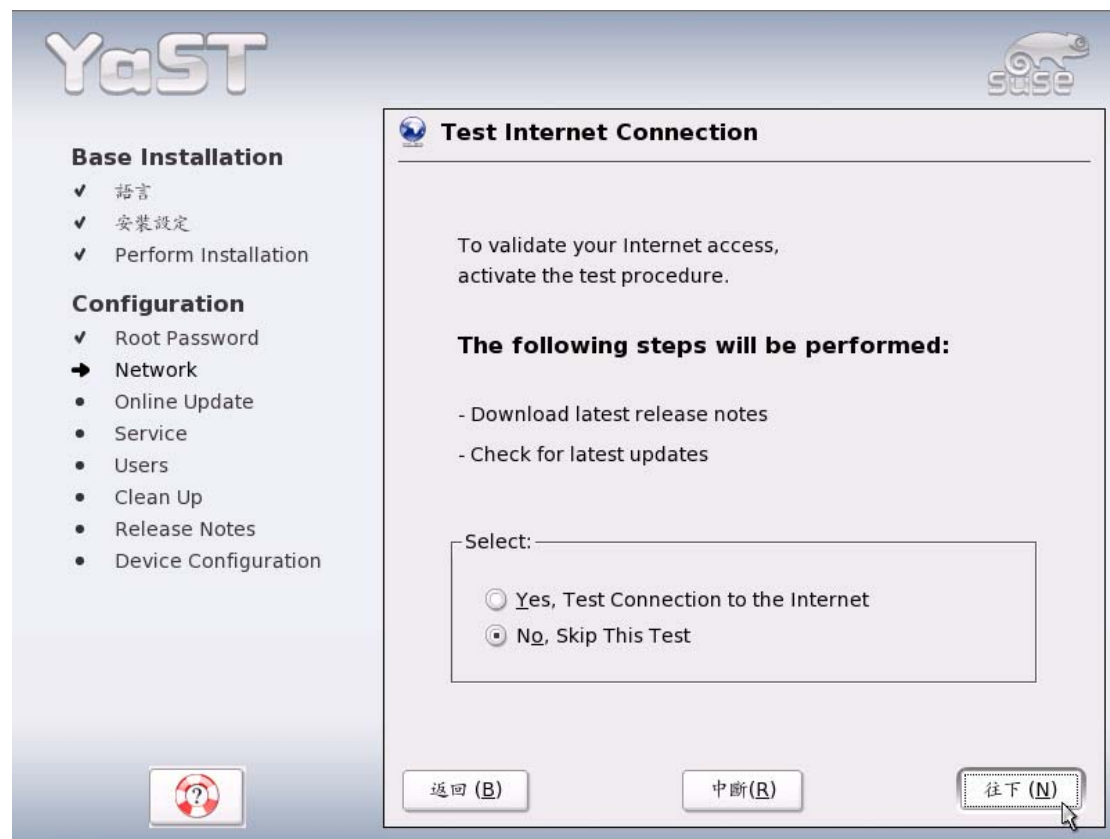


圖 1-37：測試連線畫面

步驟 11、LDAP 伺服器設定。

LDAP 可是功能強大的目錄服務伺服器，可以管理異質平台的帳號資料庫，這裡我們先採用預設值不做任何修改，如圖 1-38。讀者也可以點選「Chage...」選擇不啟動 LDAP 伺服器。

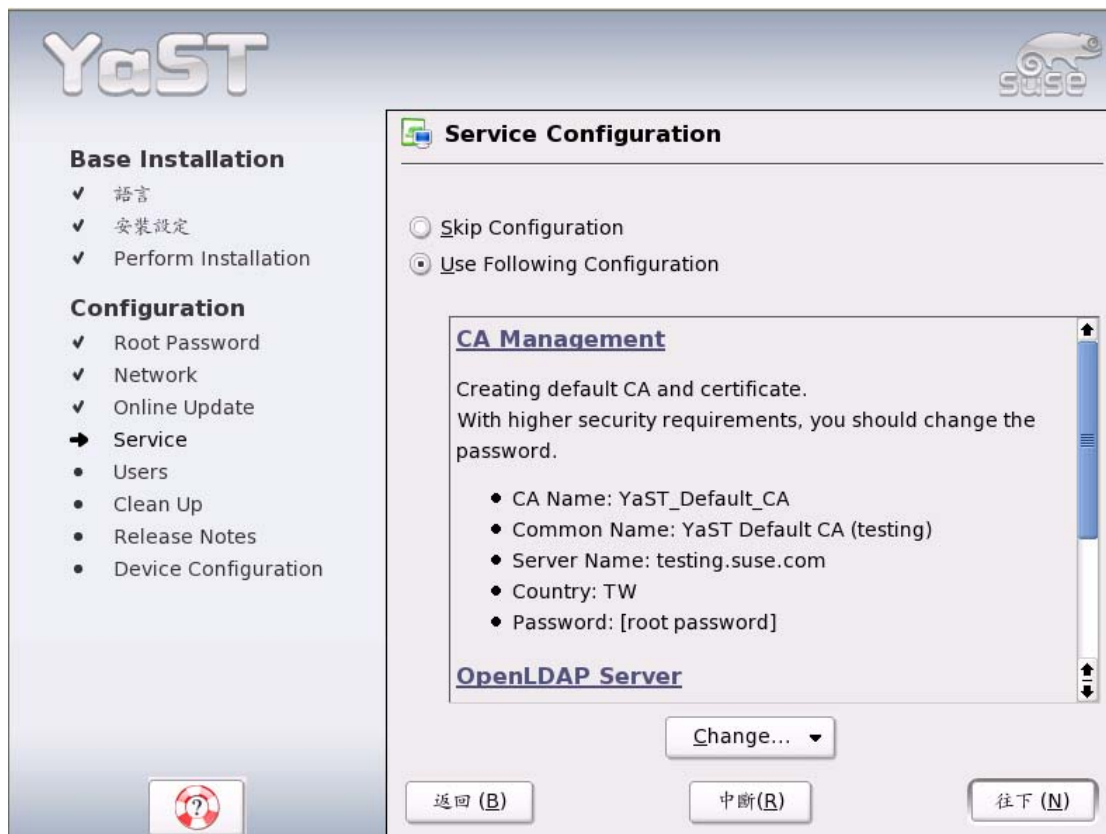


圖 1-38：LDAP 伺服器設定畫面

步驟 12、帳號管理認證方式 (User Authorization Method)。

雖然 LDAP 功能強大，不過它的設定及理論略嫌複雜，所以筆者建議使用傳統的帳號管理認證方式—「Local(/etc/passwd)」，如圖 1-39。點選「往下 (N)」繼續下一步驟。



圖 1-39：帳號管理認證方式設定畫面

步驟 13、建立一般使用者帳號。

一般使用者的帳號只具有極少的權限，但為了安全性考量，筆者建議還是要建立一般使用者，避免總是用 root 登入系統。如圖 1-40，填入使用者帳號名稱、密碼，並再次確認密碼即可。

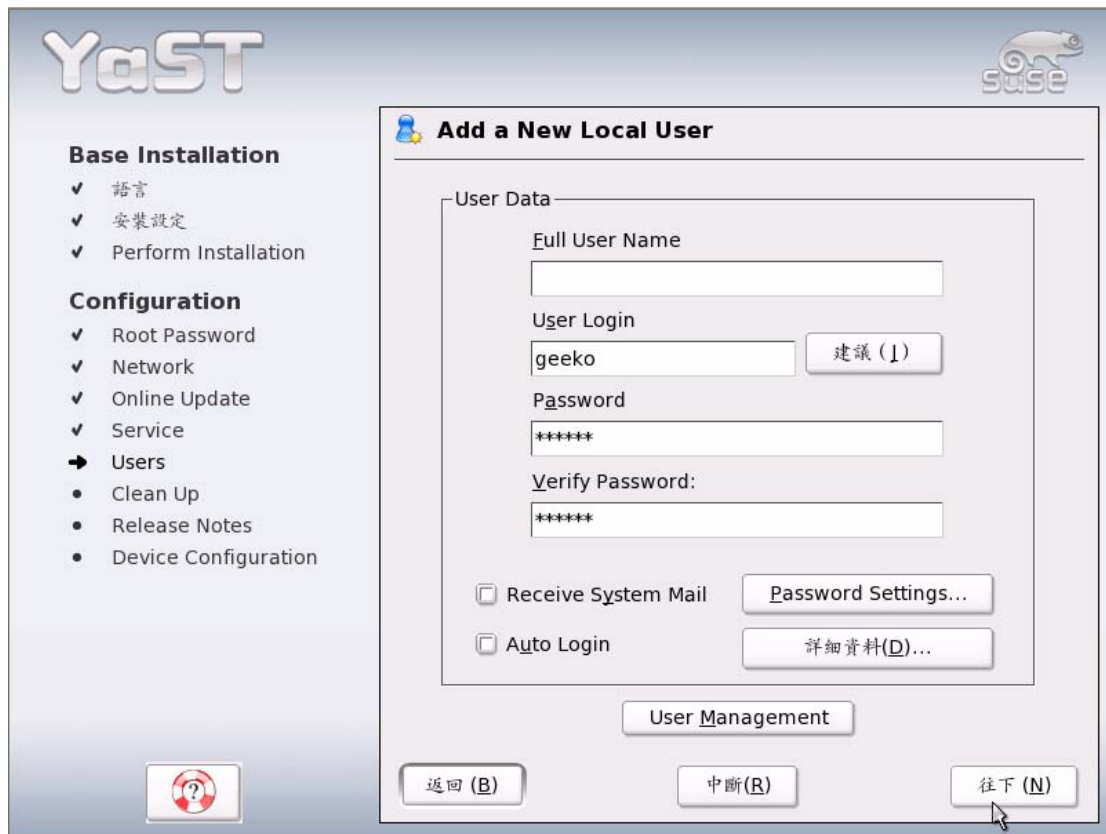


圖 1-40：建立一般使用者畫面

步驟 14、Novell SLES9 聲明畫面。

見到 Novell SLES9 的聲明畫面畫面，表示要大功告成了，快按「往下 (N)」繼續吧！

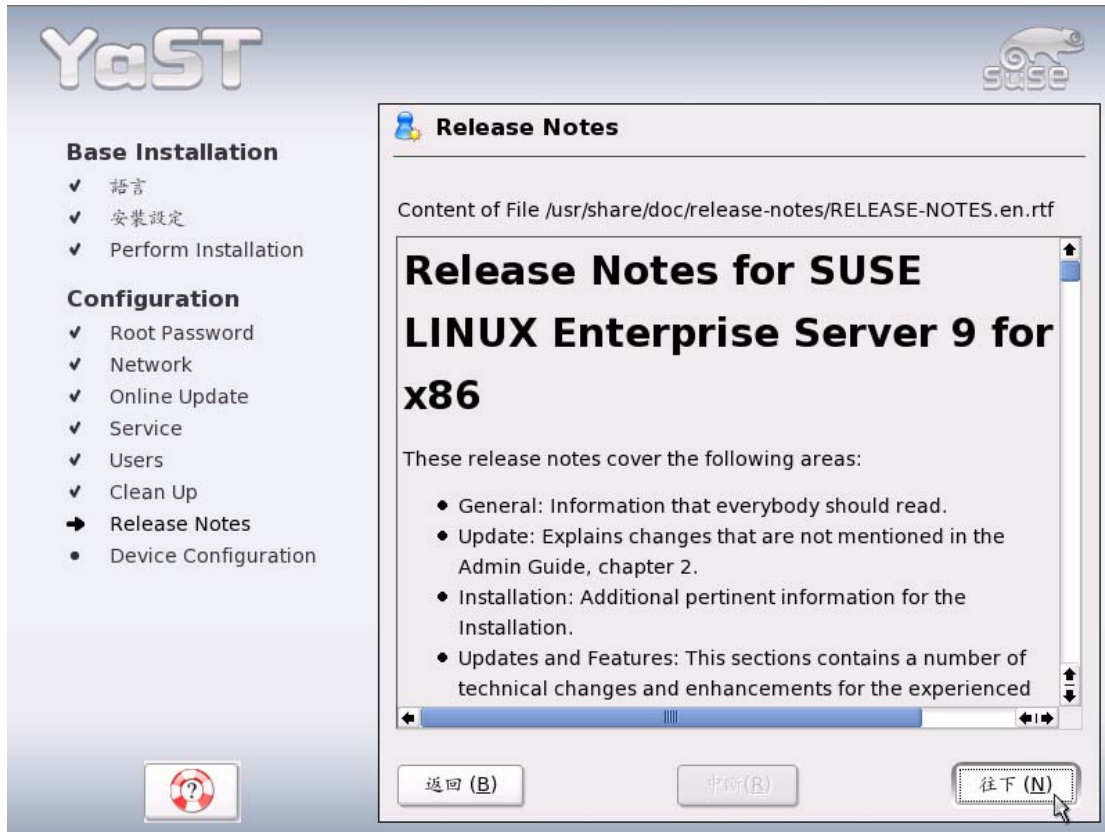


圖 1-41：版權聲明畫面

步驟 15、硬體設定畫面 (Hardware Configuration)。

通常 SLES 9 測偵到的硬體資訊應該都不會有錯，若是有任何需要異動的，請點選「Change...」進行修改，如圖 1-42。點選「往下 (N)」後，系統便會重新開機，最後出現登入畫面，如圖 1-43。輸入使用者名稱及密碼後，即可開始使用 SLES 9 的環境了！

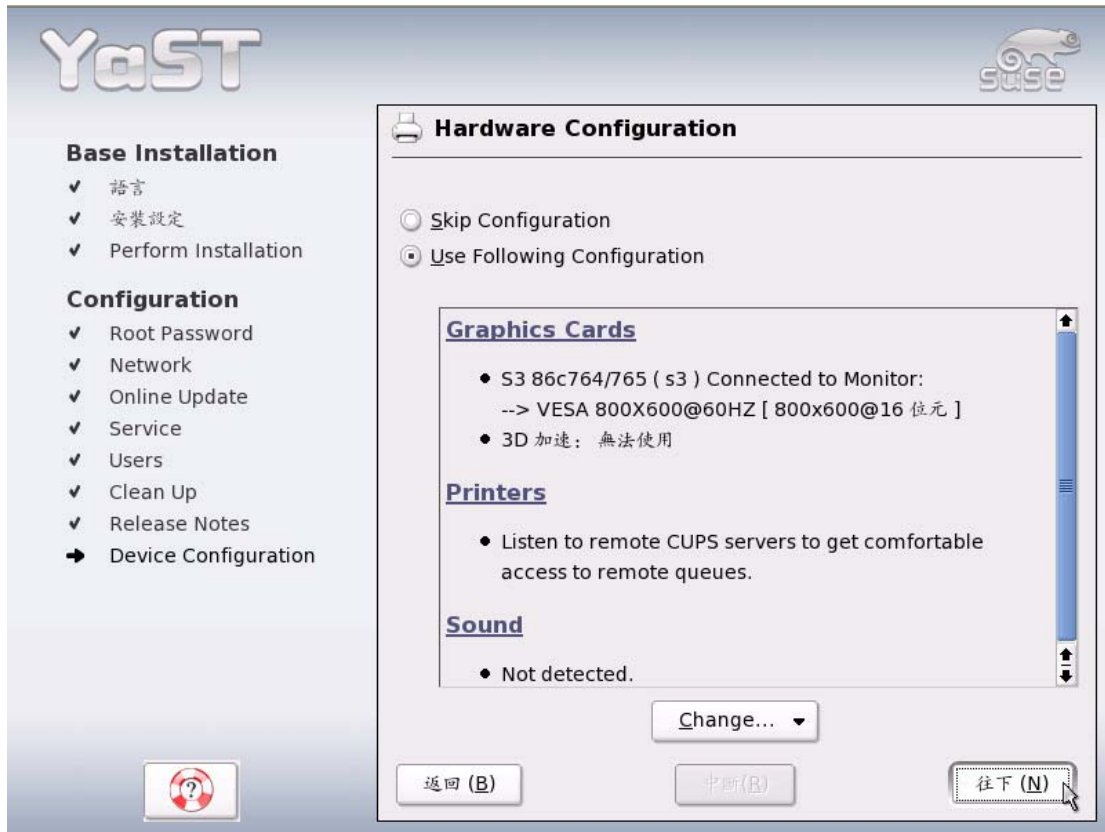


圖 1-42：硬體設定畫面

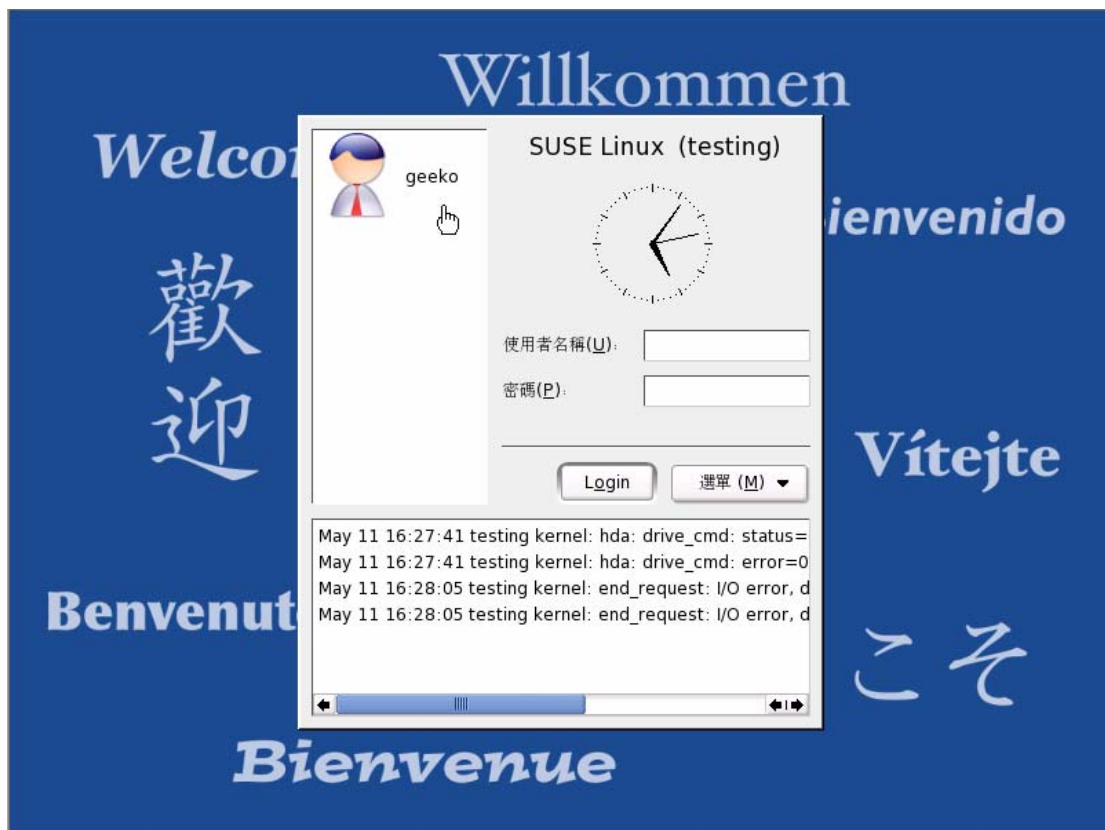


圖 1-43：SLES 9 登入畫面

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。

◆ 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。

◆ 「Primary 分割區」、「Extended 分割區」和「Logic 分割區」的關係

硬碟的分割區可分為三個種類：分別是「Primary 分割區」、「Extended 分割區」和「Logic 分割區」。

- Primary 分割區

硬碟的 partition table 的大小只有 64bytes，而每切出一個分割區需要用 16bytes 來存放相關資訊；所以一顆硬碟最多只能切出 4 個分割區，這 4 個分割區就是 Primary 分割區。

- Extended 分割區

為了突破 4 個 Primary 分割區的限制，取出其中一個 Primary 分割區（註：只能用一個）來進行擴充，這個分割區就稱為 Extended 分割區。

- Logical 分割區

Extended 分割區是無法直接使用的，必須再細分為 Logical 分割區才可用來存放資料。Logical 分割區的起始位置及結束位置記錄在每個 Logical 分割區的第一個磁區（稱為 Extended partition table）。如果要建立一個以上的 Logical 分割區，則分割程式會自動在 Extended partition Table 上產生一個分割區，如此不斷地延續下去，猶如一條鏈子，直到 Extended 分割區所涵蓋之區域用盡，故可將一部硬碟切成 N 塊區域使用。

課後練習

■ 實戰演練：VMware 下的 SLES 9

顧及讀者可能因硬體限制而無法安裝 SLES 9，所以本書附贈之光碟，內含安裝 SLES 9 所有套件的 VMware 映像檔，讓讀者無需利用額外的機器，就可體會 SLES 9 的強大功能。

步驟 1、下載 VMWare 5.0 試用版。

讀者可以到 VMWare 的網站 (<https://www.vmware.com/download/>) 取得 VMWare 5.0 試用版，請於此畫面點選 VMware Workstation 5 (for Windows operating systems and Linux systems) 的「Evaluate」，如圖 1-44。

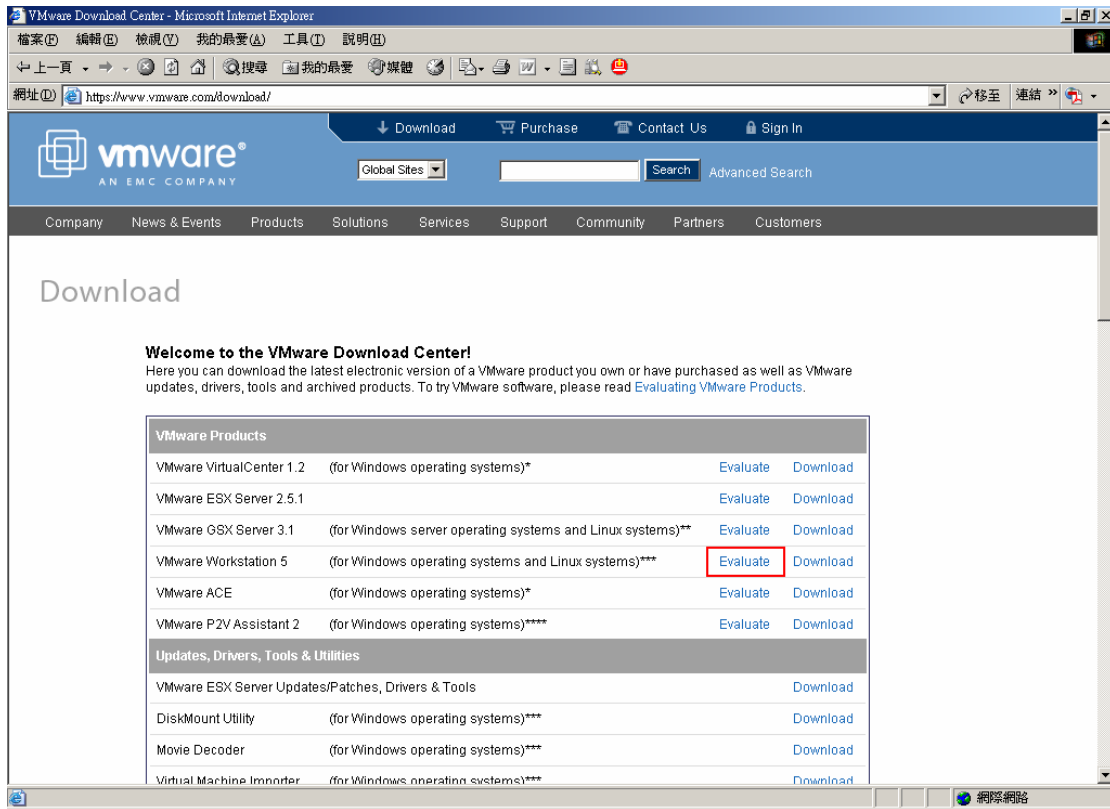


圖 1-44：VMware 5.0 試用版下載網址

接著會出現如圖 1-45 的畫面，若讀者未曾申請帳號，則需按下「Register」，隨即會出現個人基本資料的註冊畫面(圖 1-46)；確實填寫後請按下「Continue」繼續。

在出現感謝註冊畫面之後，接著會呈現版權聲明畫面，當然要按下「Yes」接受版權聲明；才能看到如圖 1-47 的「Download VMware Workstation 5.0 (or Windows Systems)」下載畫面。

Download VMware Workstation 5.0

If you've registered with VMware in the past, please enter your e-mail address and password below. If you're new to VMware, please click the Register button to continue.

Please Log in

Enter your email and VMware password to sign in using our secure server.

E-mail address:

Password:

[Forgot your password? Click here.](#)

New users

Click below to register.

圖 1-45：下載 VMware5.0 試用版網頁登入畫面

Mozilla Firefox
http://www.vmware.com/Vmwarestore/newstore/register.jsp?status=new

vmware® AN EMC COMPANY

Global Sites Search Advanced Search

Company News & Events Products Solutions Services Support Community Partners Customers

VMware Store

Register

If you've already partially registered with us, some fields may be pre-filled in.

Profile Information
Fields marked with * are required

First Name: *

Last Name: *

E-Mail Address: *

Password: *

Confirm Password: *

Company/Organization: *

Job Description: *

Address 1: *

完成

圖 1-46：基本資料填寫畫面

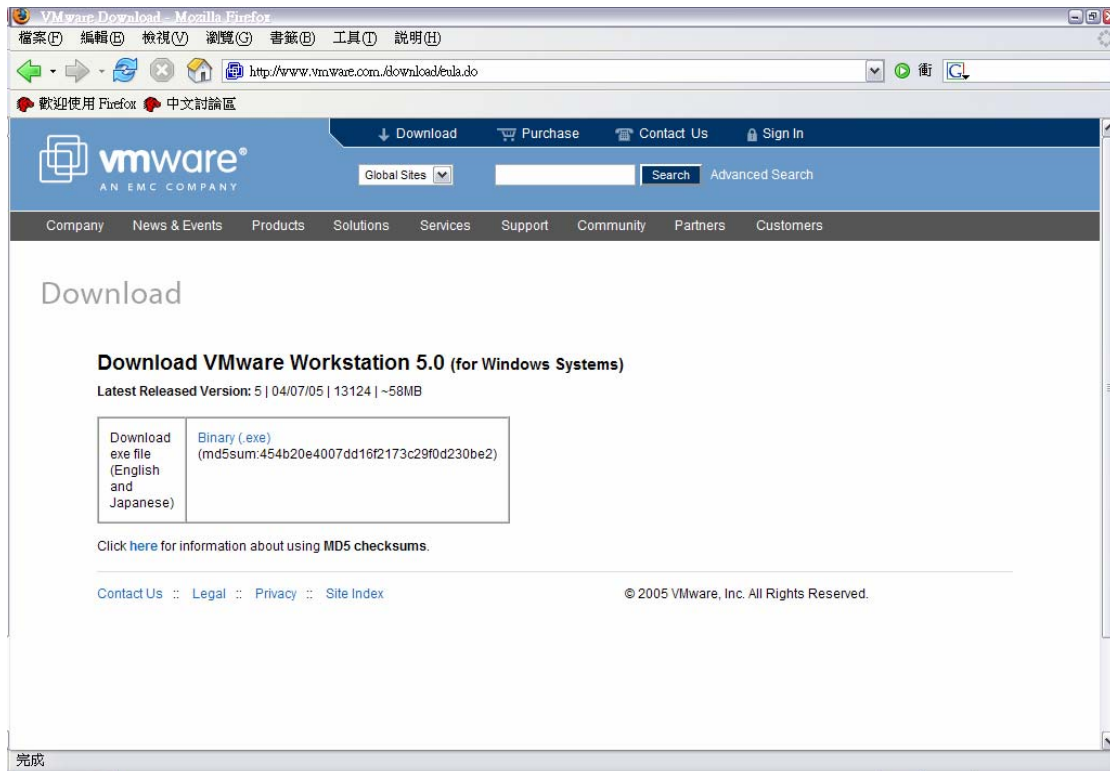


圖 1-47：下載 VMware 5.0 試用版

步驟 2、安裝 VMware 5.0 試用版。

在安裝 Windows 版的 VMware 5.0 時，幾乎是只要看到「Next >」的按鈕，點下去就行啦！跟著安裝精靈一步步的指引，應該可以順利安裝成功。接著檢查註冊時填寫的 E-mail，應該會收到類似底下由「support@vmware.com」寄來的序號。

Dear Yeh Ming,

Thank you for your interest in evaluating VMware(TM) Workstation 5 (for Windows Operating Systems). We are pleased to provide you with a **30-day** evaluation serial number.

Your evaluation serial number to use when installing VMware Workstation 5 is:

28K01-HC8A2-N8MC4-4R488

Please download the Workstation 5 (for Windows Operating Systems) evaluation software from our Web site at:

For Windows

<http://www.vmware.com/download/download.do?downloadGroup=WKST-5-WIN>

For Linux

<http://www.vmware.com/download/download.do?downloadGroup=WKST-5-LINUX>

安裝完後，當然是迫不及待地執行 VMware 了！點選上方工具列「Help」→「Enter Serial Number」（圖 1-48），便可出現如圖 1-49 輸入產品序號的畫面，填入序號便可正式啟用 VMware 5.0，不過只能試用 30 天。

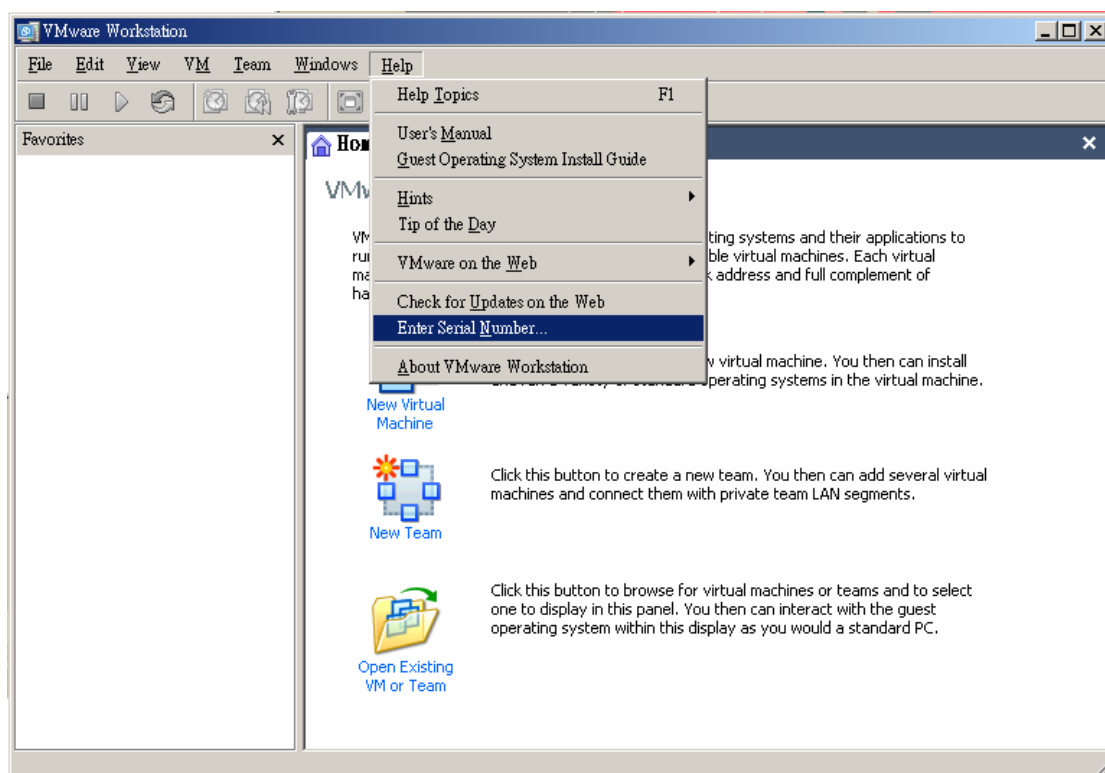


圖 1-48：輸入 VMware 5.0 序號

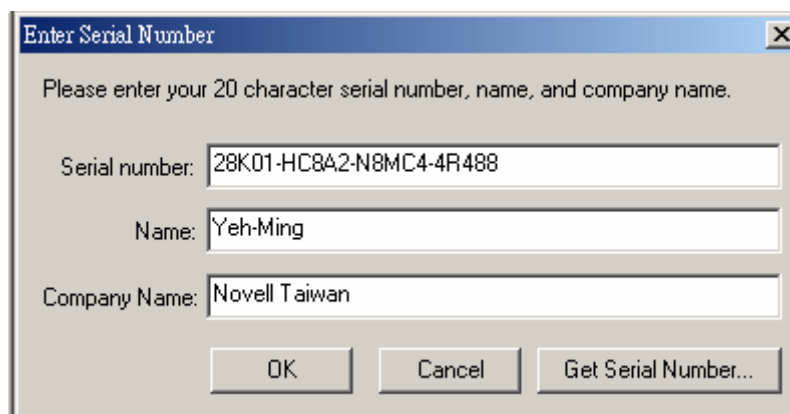


圖 1-49：VMware 5.0 序號輸入畫面

步驟 3、啟用 SLES 9 VMware 映像檔。

執行光碟片中的「testing.suse.com.exe」，並將 VMware 映像檔解至適當的位置；接著就可以點選 VMware 上方工具列中「File」→「Open...」，然後選擇 SLES 9 映像檔所在位置（圖 1-50）。

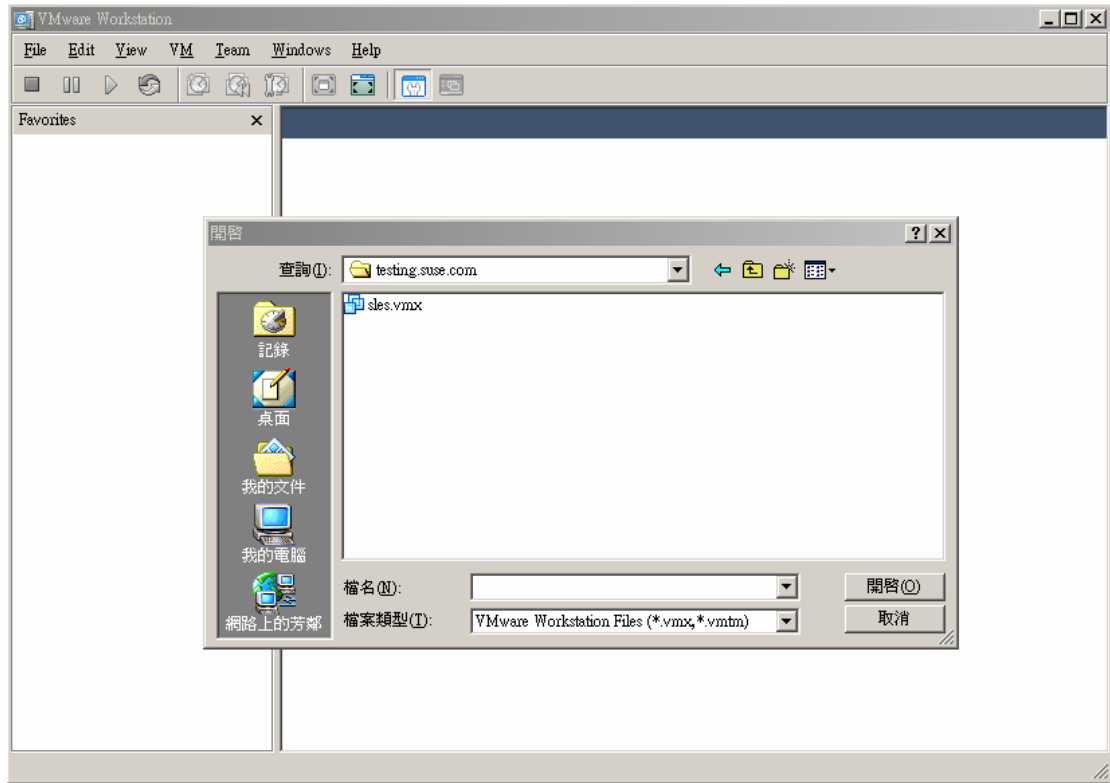


圖 1-50：開始 SLES 9 VMware 映像檔

如圖 1-51，點選上方綠色箭頭的「Power On」按鈕，便可開始啟動 SLES 9，最後會出現如圖 1-52 的 SLES 9 VMware 登入畫面。

註：光碟中所安裝的 SLES 9 VMware 映像檔，設定的登入資訊如下：

root 帳號的密碼：novell（全小寫）

一般使用者 geeko 帳號的密碼：geeko（全小寫）

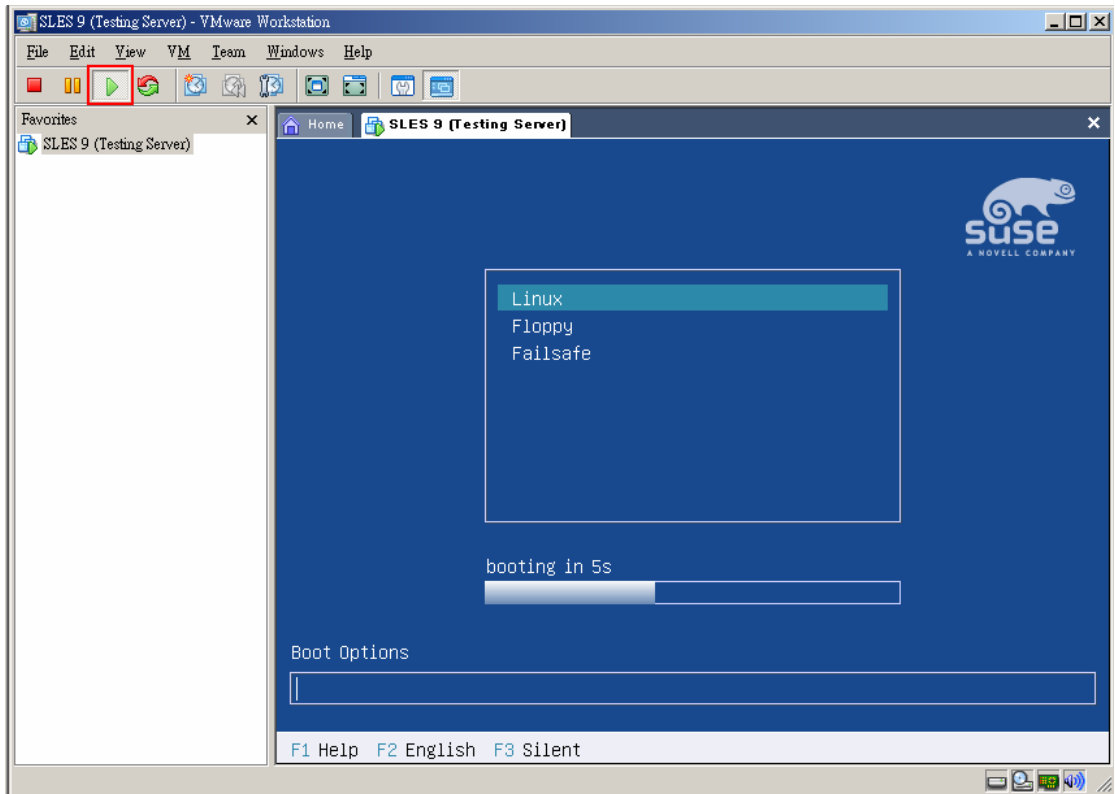


圖 1-51：啟動 SLES 9 VMware 映像檔



圖 1-52：SLES 9 VMware 登入畫面

步驟 4、SLES 9 初體驗。

讀者可以用 root 帳號登入系統；不過因為 root 的權限太大，Novell 可不建議使用者直接用 root 帳號登入，所以在登入畫面沒有看到 root 帳號。讀者還是可以直接輸入帳號「root」，密碼「novell」，以 root 身份登入並顯示如圖 1-53 的 root 桌面環境。看到桌面上一堆炸彈了吧！這就是要提醒使用者，以 root 身份來管理系統時務必要小心。若是用一般使用者帳號「geeko」，密碼「geeko」登入，則可看到如圖 1-54 的畫面。

有關 VMware 的操作，請讀者先熟悉「游標從 SLES 9 虛擬主機切換到 Windows」的快速鍵（Hot Key）為「**Ctrl+Alt**」；其他的相關設定，例如網路設定、調整虛擬主機記憶體...等，筆者在相關章節的課後練習會再做介紹。

註：在 VMware 中，安裝 VMware 的作業系統稱為「Host O.S」，以此實例便是 Windows，而寄生在 VMware 中的作業系統稱為「Guest O.S」；「**Ctrl+Alt**」便是將游標從「Guest O.S」切換回「Host O.S」的快速鍵（Hot Key）。若要再將游標切回「Guest O.S」，只需將游標移回「Guest O.S」的畫面，然後按下滑鼠左鍵即可。

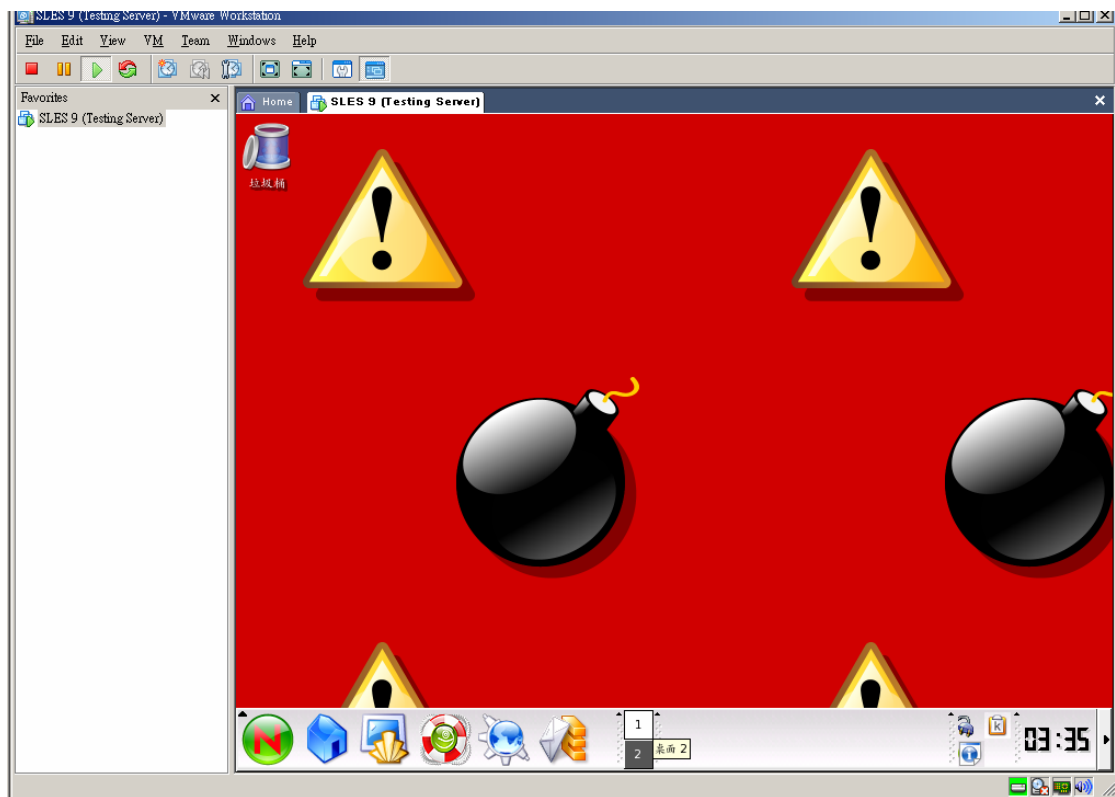


圖 1-53：root 桌面環境

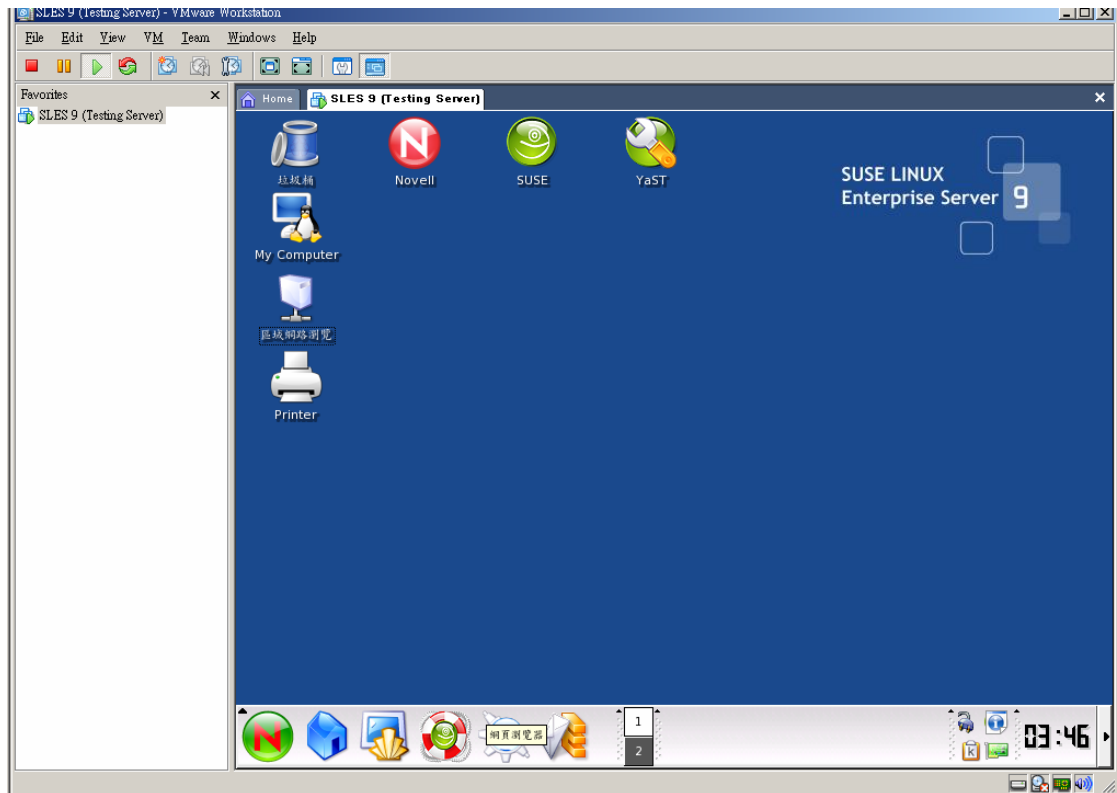


圖 1-54：一般使用者的桌面環境

葉明有感：

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！』