

次世代

Linux



隨書附贈Live DVD，免安裝，開機即可立即體驗。內容最豐富完整的Ubuntu Linux使用手冊。Ubuntu台灣社群執筆，內容最詳實

Ubuntu 玩全手冊



李聖璋 · 蔡東邦 著 上奇

EB440

次世代Linux-Ubuntu玩全手冊

出版登記證：局版北市業字第1361號
Copyright ©2006 by GrandTech C.G. Systems Inc.
上奇科技股份有限公司版權所有。

國家圖書館出版品預行編目資料

次世代Linux - Ubuntu玩全手冊 / 李聖璋, 蔡東邦著. -- 臺北市
：上奇科技，2006〔民95〕
面；公分

ISBN 978-986-7199-97-3 (平裝)

1. 作業系統

312.954

95022572

作 者 黃彬華
發行人 章孝祺
企劃編輯 莊吳行世
執行編輯 莊吳行世
發行所 上奇科技股份有限公司
地 址 台北市內湖區瑞光路76巷33號3樓
TEL : (02)8792-3001 FAX : (02)8792-3002
印刷所 士達印刷股份有限公司 2007年1月

著作權聲明 本書所有內容未經公司書面同意，不得以任何方式（包括儲存於資料庫或任何存取系統）翻譯、抄襲或節錄。

商標聲明 書中引用之商標及商品名稱，其權利分屬各公司所有，絕無侵權之意，特此聲明。

版權所有・翻印必究

目錄

推薦序

作者序－李聖璋

作者序－蔡東邦

第 1 章 Ubuntu GNU/Linux

1.1 開山祖師－Debian.....	1-2
1.2 承先啟後，Ubuntu.....	1-4
1.2.1 Ubuntu 的創始.....	1-4
1.2.2 Ubuntu 組件庫的哲學.....	1-7
1.2.3 各種不同的 Ubuntu 分支.....	1-9
1.2.4 Ubuntu 支援的不同硬體架構.....	1-11
1.2.5 Ubuntu 體貼的兩種安裝方式.....	1-12
1.3 GNU/Linux 學習資源.....	1-13
1.3.1 解決問題的心態與方法.....	1-13
1.3.2 網站.....	1-15
1.3.3 IRC 即時聊天室.....	1-17
1.3.4 如何提報錯誤.....	1-18
1.3.5 聯絡筆者.....	1-19

第 2 章 安裝前準備

2.1 硬體需求.....	2-1
2.1.1 主機硬體.....	2-2

2.1.2 支援 OpenGL 加速的顯示卡	2-4
2.1.3 目前原生支援的無線網路卡	2-5
2.2 取得光碟.....	2-9
2.3 安裝前規劃系統的預備知識.....	2-11
2.3.1 GNU/Linux 目錄的架構	2-12
2.3.2 硬體設備的代號.....	2-14
2.3.3 分割區簡介 (Partition)	2-15
2.3.4 檔案系統格式簡介	2-18
2.3.5 開始規劃你的硬碟	2-20

第 3 章 開始安裝 Ubuntu

3.1 用光碟開機	3-1
3.2 多重開機的安裝技巧.....	3-1
3.3 開始安裝.....	3-2
3.3.1 Live CD 模式安裝 Ubuntu	3-3
3.3.2 文字模式安裝 Ubuntu	3-13

第 4 章 Ubuntu & Unix Like 基本知識

4.1 絕對路徑與相對路徑	4-2
4.2 遠端登入.....	4-4
4.3 登入與登出	4-5
4.4 認識登入後的環境	4-5
4.5 Linux 的基本工具	4-7
4.5.1 檔案與目錄管理.....	4-8
4.5.2 打包與壓縮.....	4-20
4.5.3 更改檔案權限	4-27
4.5.4 尋找檔案與內容.....	4-35
4.6 萬用字元與正規表示式	4-44

4.6.1 萬用字元	4-44
4.6.2 正規表示式	4-46
4.7 vi 編輯器	4-47
4.7.1 vi 模式介紹	4-48
4.8 帳號管理	4-51
4.8.1 背景知識	4-51
4.8.2 基本指令操作	4-52
4.9 行程管理	4-55
4.9.1 前景與背景工作	4-55
4.9.2 基本指令操作	4-57

第 5 章 套件管理系統

5.1 DPKG 套件管理程式	5-2
5.1.1 deb 套件檔格式解說	5-3
5.1.2 DPKG 查詢套件	5-4
5.1.3 DPKG 安裝套件	5-9
5.1.4 DPKG 移除套件	5-13
5.1.5 DPKG 與 RPM 共舞	5-14
5.1.6 其他常用的功能	5-15
5.2 修改 /etc/apt/sources.list 設定檔來增加新的套件源	5-16
5.3 APT (Advanced Package Tool)	5-19
5.3.1 apt-cache 查詢可用套件	5-19
5.3.2 apt-get 套件升級，安裝和移除	5-25
5.3.3 apt-file 以檔案找套件	5-29
5.3.4 apt-key 金鑰管理	5-32
5.3.5 系統升級與更動	5-39
5.4 視窗模式的套件管理程式 Synaptic	5-42
5.5 從原始碼建立套件	5-45
5.5.1 使用 apt-get 來從原始碼建立套件	5-46

5.5.2 使用 apt-build 來建立最佳化套件	5-49
5.5.3 使用 auto-apt 來自動滿足 tarball 相依性	5-51
5.5.4 使用 checkinstall 來從 tarball 建立套件	5-52

第 6 章 系統設定與效能調校

6.1 網路設定	6-1
6.1.1 網路概論	6-2
6.1.2 使用圖型介面來設定有線網路	6-12
6.1.3 使用圖型介面來設定無線網路	6-15
6.1.4 使用指令列模式設定 ADSL PPPoE 撥接	6-18
6.1.5 常用的網路相關指令	6-22
6.1.6 查詢主機目前的連線	6-25
6.1.7 使用傳統設定檔的方式設定有線網路	6-30
6.1.8 使用傳統設定檔的方式設定無線網路	6-35
6.1.9 最佳化網路卡 MTU 與關掉 IPv6	6-37
6.2 安裝符合你平台的核心與顯示卡 – OpenGL 驅動設定	6-39
6.2.1 選擇適合你平台的核心	6-40
6.2.2 Nvidia 顯示卡驅動程式安裝	6-41
6.2.3 AMD (ATI) 顯示卡驅動程式安裝	6-44
6.3 印表機的設定	6-44
6.4 調整終端機 (terminal)	6-46
6.4.1 設定在視窗底下的 GNOME 終端機	6-46
6.4.2 設定 shell 顏色	6-48
6.4.3 設定 shell 底下常用指令縮寫	6-49
6.4.4 設定 framebuffer 解析度和底下的中文終端機	6-50
6.5 調整硬碟與光碟效能 hdparm	6-52

第 7 章 視窗環境

7.1 X Server 概要	7-1
7.2 各種不同有趣的 X Window Manager	7-2
7.2.1 如何在 GDM 切換不同 X Window Manager	7-18
7.2.2 安裝 KDE	7-20
7.2.3 安裝 XFCE	7-21
7.2.4 安裝 Enlightenment	7-22
7.3 認識 GNOME 基本操作	7-23
7.3.1 修改登入畫面 (GDM)	7-28
7.3.2 修改登入過程畫面 (Splash Screen)	7-33
7.3.3 修改佈景主題	7-36
7.3.4 修改系統圖示	7-42
7.3.5 修改桌面背景	7-44
7.3.6 安裝桌面小程式 (gDesklets)	7-46
7.4 中文輸入法安裝設定	7-48
7.4.1 什麼是 im-switch	7-48
7.4.2 scim 萬國語言輸入程式	7-49
7.4.3 gcin 中文輸入程式	7-51
7.4.4 gcin 之嘸蝦米輸入法	7-52

第 8 章 網路應用程式

8.1 網頁瀏覽器 FireFox	8-2
8.1.1 Flash Player 安裝	8-3
8.1.2 JAVA Runtime Environment 安裝	8-4
8.1.3 Adobe Reader PDF 閱讀程式外掛	8-5
8.1.4 Mplayer 多媒體播放器外掛	8-7
8.1.5 Totem Gstreamer 多媒體播放器外掛	8-8
8.1.6 Totem xine 多媒體播放外掛	8-9

8.1.7 一些好用的外掛.....	8-10
8.2 E-Mail 收信軟體.....	8-17
8.2.1 Thunderbird (包含 RSS feed)	8-18
8.3 即時通訊軟體.....	8-22
8.3.1 Gaim 支援最多通訊協定的即時通訊軟體.....	8-23
8.3.2 skype 目前網路電話的領先者.....	8-28
8.3.3 Ekiga 支援 SIP 協定的網路視訊電話.....	8-33
8.4 網路聊天程式.....	8-43
8.4.1 PCManX – 自由的 BBS 連線軟體.....	8-43
8.4.2 Xchat – 可連上高手眾多的 IRC 聊天室.....	8-45
8.5 檔案傳輸軟體.....	8-49
8.5.1 Linux 上老牌的 FTP 軟體 – gFTP.....	8-49
8.5.2 FTP 的另外一個好選擇 – Filezilla.....	8-50
8.5.3 Linux 上的 FlashGet – Downloader for X.....	8-54
8.6 P2P 檔案分享.....	8-56
8.6.1 好用的 BT 下載程式 – Azureus.....	8-56
8.6.2 Linux 下的騎驢 (eMule) 程式 – aMule.....	8-61

第 9 章 多媒體影音欣賞

9.1 一些額外的多媒體支援安裝.....	9-1
9.1.1 破解萬惡的 DVD 影片區碼.....	9-1
9.1.2 安裝 Win32 Codecs 影片解碼器.....	9-3
9.2 Mplayer – Linux 中支援最多格式的軟體播放器.....	9-4
9.2.1 使用 Mplayer 播放有字幕檔的 avi 影片.....	9-4
9.2.2 使用 Mplayer 播放 DVD VCD.....	9-7
9.3 XINE – 介面類似 PowerDVD 的多媒體播放器.....	9-9
9.3.1 使用 XINE 播放有字幕檔的 avi 影片.....	9-10
9.3.2 使用 XINE 播放 DVD 和 VCD.....	9-11
9.3.3 Gxine – Gnome 介面的 xine 播放器.....	9-12

9.4 VLC – 跨平台的多媒體播放器.....	9-14
9.4.1 自動偵測字幕檔.....	9-14
9.4.2 設定自型檔和字體大小.....	9-14
9.4.3 選擇字幕格式.....	9-15
9.4.4 選擇字幕編碼.....	9-15
9.5 Totem – 預設的多媒體播放器.....	9-16
9.5.1 使用 Gstreamer 當播放引擎.....	9-16
9.5.2 使用 XINE 當播放引擎.....	9-19
9.6 RealPlayer.....	9-20
9.7 beep-media-player 源自於 xmms 的 mp3 播放器.....	9-23
9.8 Rhythmbox 收錄 CD 控制 ipod.....	9-25
9.9 amaroK.....	9-28
9.10 廣播收聽軟體 – Directory Browser (streamtuner).....	9-30
9.11 用 Linux 看數位電視.....	9-32

第 10 章 一些實用的應用軟體

10.1 圖片瀏覽與編輯程式.....	10-1
10.1.1 Linux 中的 photoshop – Gimp.....	10-1
10.1.2 輕便快速的圖形瀏覽程式 – GQview.....	10-10
10.1.3 簡易的影像檢視器 – Gwenview.....	10-13
10.1.4 好用的圖片管理程式 – gThumb.....	10-15
10.1.5 預設的圖片管理程式 – Eye of GNOME.....	10-18
10.2 StarDict (星際譯王) – 好用的翻譯軟體.....	10-19
10.2.1 安裝不同的字典檔.....	10-19
10.2.2 安裝人聲發音檔.....	10-21
10.3 燒錄軟體.....	10-22
10.3.1 K3b – KDE 底下的燒錄軟體.....	10-22
10.3.2 GnomeBaker – Gnome 底下新的燒錄軟體.....	10-24
10.4 文書處理軟體.....	10-25

10.4.1 文件製作軟體－OpenOffice.org Writer	10-25
10.4.2 試算表製作軟體－OpenOffice.org Calc	10-26
10.4.3 簡報製作軟體－OpenOffice.org Impress	10-27
10.4.4 流程圖與圖形繪製－OpenOffice.org Draw	10-29
10.4.5 方程式編輯－OpenOffice.org Math	10-30
10.5 PDF 瀏覽程式	10-32
10.5.1 Adobe Reader 7	10-32
10.5.2 Evince－內建的 PDF 瀏覽程式	10-33
10.5.3 Kpdf	10-34
10.5.4 Xpdf	10-35
10.6 網頁製作軟體－Nvu	10-35
10.7 PCMan File Manager	10-36
10.8 Lightweight X11 Desktop Environment	10-38

第 11 章 用 Wine 來執行 Windows 的 遊戲與程式

11.1 wine 概觀	11-1
11.2 裝上 IE	11-2
11.2.1 IEs 4 Linux	11-2
11.2.2 下載 IEs 4 Linux	11-3
11.3 用 Wine 來裝上你的魔獸世界	11-6
11.3.1 編譯你專屬的 Wine	11-7
11.3.2 建立 .wine 資料夾	11-8
11.3.3 下載安裝魔獸世界所需要的 dll 檔	11-9
11.3.4 開始安裝魔獸世界	11-11
11.3.5 魔獸世界安裝程式	11-12
11.3.6 設定以 OpenGL 開啟遊戲	11-15
11.3.7 音效設定	11-15
11.3.8 3D 加速	11-15
11.3.9 啟動魔獸世界	11-16

11.4 用 Wine 安裝 Steam	11-21
11.4.1 Steam 主程式下載	11-21

第 12 章 Linux 遊樂場

12.1 PPracer, 3D 企鵝滑雪	12-2
12.2 Tuxkart, 3D 的小企鵝賽車	12-3
12.3 Supertux, 超級瑪俐? 超級企鵝!	12-3
12.4 Armagetron, 類似貪食蛇的 3D 線上遊戲	12-4
12.5 bzflag, 3D 線上戰車遊戲	12-5
12.6 Tremulous, 3D Online FPS	12-6
12.6.1 Tremulous 簡介	12-6
12.6.2 安裝 Tremulous	12-7
12.7 Enemy Territory, 網路 FPS 遊戲	12-10

第 13 章 深入瞭解 Ubuntu 系統

13.1 開機流程	13-1
13.2 fstab 開機磁碟掛載與 UUID 新架構	13-2
13.2.1 為何引入 UUID 與新的管理方式	13-3
13.2.2 使用 NTFS-3G 讀寫 NTFS 分割區	13-4
13.3 Runlevel 原理和設定	13-6
13.3.1 init.d 內的各種伺服器控制檔	13-8
13.3.2 inittab 預設 runlevel 層級	13-9
13.3.3 管理某服務是否開機後啟用	13-12

第 14 章 網頁伺服器 (LAMP 架構)

14.1 LAMP 網頁伺服器運作原理	14-2
14.2 LAMP 安裝架設	14-3
14.2.1 Apache2 安裝	14-4

14.2.2 PHP 安裝	14-5
14.2.3 MySQL 安裝	14-7
14.3 Apache2 設定	14-8
14.3.1 系統架構	14-8
14.3.2 apache2.conf 設定	14-14
14.3.3 網站設定檔	14-15
14.3.4 增加虛擬主機	14-17
14.3.5 使用 mod_deflate 來壓縮網頁並節省頻寬	14-18
14.3.6 啟用個人網頁	14-22
14.4 PHP 設定	14-23
14.4.1 設定檔設定	14-23
14.4.2 使用 eAccelerator 來加速 PHP 程式	14-24
14.5 MySQL 設定	14-28
14.5.1 忘了或修改 MySQL 密碼	14-28
14.5.2 資料庫備份	14-30
14.5.3 開啟 port 3306 讓非本地端主機連線	14-32

第 15 章 DHCP 伺服器

15.1 DHCP 的運作理論	15-2
15.2 DHCP 實際安裝	15-3
15.2.1 發放浮動式 IP	15-4
15.2.1 發放固定式 IP	15-6

第 16 章 Samba Server 與 Windows 互相分享檔案

16.1 用圖型介面來裝上 Samba server	16-2
16.2 設定使用者帳號密碼	16-6
16.3 用 Ubuntu 進入 Windows 網芳鄰	16-8
16.4 SAMBA 的相關設定	16-9

16.5 Samba Server 的啟動與關閉.....	16-11
-------------------------------	-------

第 17 章 NFS 伺服器

17.1 NFS 運作理論.....	17-2
17.2 NFS Server 主機規劃.....	17-4
17.3 NFS Server 和 Client 程式安裝.....	17-6
17.3.1 NFSv3 Server 的設定（主流穩定版）.....	17-7
17.3.2 NFSv4 Server 的設定（實驗性質）.....	17-11
17.3.3 將設定寫到 fstab 來自動掛載吧！.....	17-14

第 18 章 FTP 伺服器

18.1 基礎 FTP 概念.....	18-1
18.2 主動（Active）連線和被動（Passive）連線.....	18-3
18.3 FTP 伺服器.....	18-4
18.3.1 安裝、登入 FTP 伺服器.....	18-5
18.4 ProFTPD 主要設定.....	18-11
18.5 UTF8 和 Big5.....	18-13

第 19 章 圖型化界面的防火牆與 iptables 概觀

19.1 Linux 防火牆功能.....	19-1
19.2 快速建立防火牆 – Firestarter.....	19-3

第 20 章 Beryl 3D Desktop

20.1 Beryl 是什麼？.....	20-1
20.2 安裝 Beryl.....	20-2
20.2.1 增加第三方套件庫來源.....	20-2
20.2.2 安裝顯示卡驅動程式.....	20-3
20.2.3 安裝 Beryl.....	20-4

20.2.4 修改 xorg.conf.....	20-4
20.2.5 啟動 Beryl X Window Manager	20-6
20.2.6 將 Beryl 加到作業階段	20-13

第 21 章 Edgy 新增加的程式

21.1 Tomboy 好用的筆記本工具	21-1
21.2 F-Spot Photo Manager 相簿管理程式	21-2

推薦序

享受使用 Linux 的興趣

眾所皆知的，Linux 的發行版本相當的多，諸如 Red Hat 系統、SuSE、Mandriva 與 Debian 等等。在這些常見的發行版本當中以 Debian 這個社群版本被譽為架構最為嚴謹的 Linux 發行版本，更有朋友說過「不曾玩過 Debian 版本的人不能稱為 Linux 高手」的言論呢！^_^！不過由於 Debian 是由社群的高手們維護的，雖然體質相當優良，不過學習進入的門檻相對的比較大一些。

Ubuntu 是近年來才推出的一個 Linux 發行版本，這個版本主要建構在 Debian 上面，因此先天上具有相當良好的體質，加上 Ubuntu 本身的創意不少，讓 Ubuntu 得以在 Linux 桌面使用上佔有一席之地。除了桌面軟體整合度相當高之外，Ubuntu 在傳統的 Linux 伺服器設定上面也作得有聲有色，也是一套相當棒的 Linux 發行版本，很值得使用者一探究竟的。

由東邦兄與聖瑋兄等合撰的這本《次世代 Linux—Ubuntu 玩全手冊》，其內容可以分為幾個部分：

第一部份主要在 Ubuntu 的內容介紹與安裝、套件管理與效能調校、X 桌面的使用與調校等等。這部份介紹了詳細的圖解安裝，以及初次接觸 Linux 的朋友最害怕的驅動程式是否成功啟動等問題。在 X 桌面上則介紹了大多數辦公軟體，以及常用的上網工具等等，讀完這一部份，這部 Ubuntu 的 Linux 系統已經足以當成您的辦公用主機囉！

第二部分主要在介紹如何在 Linux 的環境下模擬 Windows 的環境以及很多相當有趣的 Linux 遊戲軟體。目前將 Linux 做為桌上型電腦的朋友最麻煩的地方可能在於「那我原來的遊戲能不能繼續玩？」這個部分可以透過 Wine 這個可以在 X 環境下執行 Windows 相容程式的軟體來處理呢！此部分看完後，讀者們應該就能夠在 Linux 這個相當棒的作業系統上面玩您之前在 Windows 上才能跑的遊戲囉！

Linux 最棒的也是最常用的功能就是網路伺服器囉！所以作者在第三部分介紹了幾個常見且熱門的伺服器，包括 WWW 伺服器、NFS 伺服器等等，雖然礙於篇幅該伺服器介紹是點到為止，不過其中某些特殊且重要的設定參數，例如 Apache 伺服器內的 AddDefaultCharset 的參數等，作者有很詳細的加以說明，而不是單純講解該參數的意義而已。所以看完此部分相信讀者對於伺服器會有更深一層的認識。

本書沒有艱深的理論說明，沒有複雜的指令介紹，由相當年輕的作者群針對 Ubuntu 這個 Linux 發行版本的特色來介紹常見的 X 視窗軟體以及伺服器功能，內容相當符合大眾閱讀，相信您讀完本書後會對 Ubuntu 這個發行版本有更深一層的體認。不囉唆，趕緊享受閱讀的樂趣吧！

鳥哥
2006/11 台南

大家一起來玩 Ubuntu 吧！

一直以來，Linux 都給人一種不易上手，只適合高手和駭客使用的刻板印象。在許多年前，這的確曾是事實，但經過多年發展之後，今天的 Linux，已經有了全新的面貌，各種發行版本不但使用介面越來越友善，中文支援也日趨完善。而在今日眾多的發行版本中，最受矚目的，當屬來自南非的 Ubuntu Linux 到底是什麼樣的原因，使得這個新系統能夠急速竄紅，以黑馬之姿脫穎而出？Linux 又到底有什麼令人著迷的魅力，讓這麼多人願意花時間去學習？我想，這本書，應該可以提供給讀者一部分的答案。那另一部分呢？另一部分，當然就得要等讀者親自去體驗了。

如果您曾經有想嘗試 Linux 的念頭，卻因為不知道如何入門上手而作罷，您的機會來了。Ubuntu 本身就是非常適合初學者入門使用的 Linux 系統，學習門檻已經降低了非常多，而剩下一些比較不容易的部份，就由這本《次世代 Linux—Ubuntu 玩全手冊》來幫您補足吧！

這本書涵蓋了安裝、基本觀念介紹、基本指令、重要的系統設定、桌面環境操作、網路應用、多媒體等日常生活中相當實用的主題，而更重要的是，裡面甚至還有一個專講如何在 Linux 下玩遊戲的章節，介紹了如何在 Linux 下玩《魔獸世界》的方法，還介紹了幾款遊戲，不僅破除了 Linux 下無法玩遊戲的迷思，同時也提供了實際的操作步驟，算是本書很有特色的部份。當時作者邀我寫序時，我沒有馬上答應，而是要求一定要親自讀過，了解實際內容之後，我才能決定，確保這並不是「友情推」。在讀完作者寄給我的稿件之後，雖然有些地方還是覺得有些許改善空間，但整體來說，我還是會給予高度評價。

個人覺得本書內容相當實用，大部分使用上比較常遇到的問題大致都有涵蓋到，而基礎和應用，指令和圖形介面操作的篇幅，比例也算是恰到好處。縱使有些小瑕疵，仍然瑕不掩瑜，值得推薦，適合作為入門 Ubuntu 的導引。但在這裡要提醒讀者的是，作者在前面入門的章節，寫了許多指令操作的部份。乍看之下似乎很不好學，但實際上，現在在桌面環境下，大部分的事情已經不需要打指令，只是因為那些指令也算是重要的基礎操作和觀念，而且要是離開了桌面環境，就會需要用到，所以才會被放在前面的章節先行介紹，剛入門的讀者可別被嚇到了，那些東西，一開始並不需要全部學會，只要先稍微有個概念，很多東西往後還是可以慢慢學的。

幫一本書寫推薦序，有時就像是在寫美食介紹，作者可以天花亂墜，極盡所能描述食物是如何的美味，但若非親自嚐上一口，您永遠不會知道那是什麼滋味。而現在我要告訴您的是，Ubuntu 不但美味可口，而且還是完全免費的「正版」軟體。所以，今天就跟我們大家一起，來玩 Ubuntu 吧！

洪任諭 PCMan

Ubuntu — 您最佳的選擇之一

談到 Linux 的發展，這兩年可以說是非常豐收的。尤其是在 Desktop 方面的發展，最讓人驚喜。同時，使用 Linux 當作日常操作系統的人也越來越多。過往很多難設難玩的問題都逐漸得以解決。雖然我不喜歡拿 Linux 來跟 Windows 來比較，但我這裡很想跟大家分享一個經驗：在一次以 Linux 新鮮人為主要聽眾的演講中，一上場我就展示了我的 Linux 桌面（當然，啟動時我沒接投影），我跟台下的朋友說我用的是最新版的 Windows（那時 Vista 還沒出），居然有一半人被我騙了！ ^_^

雖然 Ubuntu 這個名字對很多人來說很陌生，但是假如您在這兩年您有留意 Linux 方面的新聞的話，那我敢保證 Ubuntu 這個名字一定出現過您的眼簾裡。它的名字背後含著甚麼意義？它究竟有甚麼能耐？它為何能成為當今炙手可熱的當紅炸子雞？它跟過往其他的 Linux 版本的淵源有甚麼瓜葛？還有，我也可以輕鬆愉快的與之共舞嗎？等等…

這些問題的答案又在哪裡呢？我想這也是很多朋友所感興趣的。

很感謝兩位作者這段時間的辛勞付出，才有了本書的出世。這剛好是很多 Linux 朋友所需的！書本的兩位年輕作者，用他們自身的經驗為讀者們展開精彩 Ubuntu 之旅：從其歷史故事、架構理念、系統管理、桌面應用、到伺服器架設，還有很多精緻的技巧心得。足以讓您從各方面好好地感受一下 Ubuntu 的震撼教育。

我答應 Ducati 幫他寫這本書的序是早幾個月之前的事情，不過等到稿子到我手上時，已經臨印前夕了。本想草草看一看交差了事的（哈，作者別打我！）。不過，當我看完書目及前兩章之後，卻讓我忍不住從頭到尾細細的品嚐起來。並且照著書本的步驟將我的 Ubuntu 升級了一遍，還安裝了不少有趣的程式，跟著書本的介紹玩了起來。就算您從來沒用過 Linux 也沒關係，Ubuntu 操作介面真的很友善！如果您沒有真的操作過，尤其是您之前使用過其他版本的 Linux 的話，Ubuntu 的種種便利性其實很容易體會。即使您從 Debian 轉過來，也能從 Ubuntu 上發現很多驚喜！這種體驗是讓人很愉快且讚賞的！

假如，您曾經認為 Linux 很難用，並且以之作為拒絕的理由。那麼，Ubuntu 會讓您覺得之前的理由將不再是藉口。

假如，您已經成為一位 Linux 的日常使用者，且一直在尋覓一套更適合您個人喜好的版本，Ubuntu 無疑是您最佳的選擇之一。

現在，該是行動的時候了！

netman

2006-10-31 於高雄到台南的火車上

做電腦的主人

雖常在網路上發表愚見，但以真名發表推薦序還是頭一遭。日前，友人 DUCATI（李聖璋）敦請小弟為這本書作序時，毫不猶豫就答應，衷心期待這第一本關於 Ubuntu GNU/Linux 的台灣中文書籍能讓讀者有所收穫。

姑且不談 GNU/Linux，試看當今軟體的發展狀況：硬體的進步是隨著知名的摩爾定律在前進，也就晶片半導體數量以每十八個月的速率倍增，其成長幅度的劇烈，對每個稍微資深的電腦使用者而言，都可明顯感受到，然而，軟體的發展卻依循 Parkinson 定律（完成一件工作所需的時間，會擴大至該工作所容許的時限），換言之，無論電腦硬體進展有什麼重大突破，總是趕不上軟體的無止盡需求。Microsoft Windows 的異軍突起、與 Intel 掛帥成為「Wintel 帝國」，可說是 Parkinson 定律與摩爾定律的最佳典範。為了撰寫推薦序，特別去電腦書店走一遭，琳瑯滿目的選擇令人心生怯意，電腦技術曾幾何時，已經複雜至斯？！

電腦的發明就是要替人類解決問題，應該是人類最忠實的僕人，然而曾幾何時，許多人們反而被電腦所奴役，在有心人的掌握與布局下，許多人們受限於既定的框架和思維方式，賦予電腦生命的軟體也趨於封閉，該如何讓電腦確切做事，而非僅是讓人們永遠去遵循那些偉大的電腦規則呢？自由軟體之父 Richard M. Stallman 正視這個問題，並籌創自由軟體基金會（Free Software Foundation），激發人類對設計電腦最初的呼喚：「人類設計軟體以操控硬體，而軟體本身應該要是自由的，這樣電腦才能作任何人們期望的工作」，而我們也見到 GNU/Linux、*BSD，以及一系列自由軟體的興起，正是人類對自由的訴求。

屈指一算，接觸 GNU/Linux 對小弟而言，斷斷續續近十年，原本只是為了中學的科學展覽，開發一套小型作業系統，參考 GNU/Linux 的核心作基礎，沒想到作業系統還沒寫好，就沉迷於 Linux 的美妙世界。作為公開討論與技術分享的 LKML (Linux Kernel Mailing-List)，時常有許多驚人的進展與創新被提出，而我也目睹 KDE 與 GNOME 桌面系統計畫被提出並落實，將這些軟體匯集的 Linux Distributions 更是琳瑯滿目，只要有不同的訴求，幾乎都可透過自由軟體的組合再造，創造新的 Linux 系統，Ubuntu GNU/Linux 就是極佳的範例。不過，GNU/Linux 最獨特之處，還是在於廣大的「社群」(community) 力量，Linux 嚴格來說只是作業系統的核心，GNU 計畫自 1985 年至今，累積大量的高質量軟體，這些交叉的組合與交互作用，激發相當驚人的效應。

本身作者之一 dbtsai (蔡東邦) 就是一位活躍於 Ubuntu 使用者社群的朋友，我與他素未謀面，但常見到他在論壇或使用者討論群組的貢獻，累積的 credit 讓我不僅認同其正面價值，更欣然見到他無私地分享在 Ubuntu GNU/Linux 的心得。這不是一本技術高深的書籍，相反地，透過平易近人的口吻，試圖將 Ubuntu 所強調的”Linux for Human Begins” 精神彰顯，我們不需要另一個龐大不堪的作業系統來奴役自我、更不想要迷思於死板的操作形式中，相反地，我們即將可見，的確可讓電腦發揮其價值，達成我們期待的行為，在這個開放且自由的系統中，我們將得以擁有桌面系統、網路伺服器、華麗的多媒體享受等等，不只能作細部微調，事實上，只要您喜歡，甚至可體驗「庖丁解牛」的美妙，面對這些自由軟體，「源碼之前了無秘密」，不僅可一窺宮廟之美，還可感受到完全「奴役」硬體的成就感。Ubuntu GNU/Linux 大大簡化邁入如此境界的難度，本身作者將帶領各位讀者，按部就班地張羅所需的背景知識、安裝基礎系統與軟體套件，從而享受種種 Ubuntu 帶來的便利性，最後沉浸於這自由且開放的國度中。

初入這自由的國度，難免有「劉姥姥逛大觀園」之嘆，儘管在搜尋引擎發展成熟的今日，仍不免有所迷思，這本書的旨意並非要讀者按圖索驥、Step-by-Step 作死板的操作，而是提供許多切入點，讓讀者得以建立清楚的觀念，稍後可參考

書中所提到的眾多資源，進一步深入探索，創造 Ubuntu GNU/Linux 的新價值，請記住讀者將是電腦的主人，而不再受死板而封閉的軟體所奴役。

本書另一位作者 DUCATI 活躍於 Ubuntu 台灣的社群圈子中，也常作經驗分享。我們生活的年代，被哲學家奧登稱為「焦慮的時代」，面對浩瀚的技術範疇，我們不再有如同蘇東坡夜遊赤壁而生「物與我皆無盡」的體悟，相反地，常常惶恐於不知如何立足，Ubuntu GNU/Linux 不僅是開放且自由的作業系統，其社群共享的精神，更是綻放美麗花蕊的催手，DUCATI 與 dbtsai 起筆這本著作，也算是此等精神的延續，當然著作只是開端，重點是知識的擴展與再生。希望讀者在閱畢後，即使無法消除對技術的瞬息萬變產生的焦慮，也能拉近與 GNU/Linux 和多元化應用的距離，進而產生共鳴，並投身於這二十年來對電腦技術的自由解放的「文藝復興」運動中。

謹此推薦本書予讀者朋友們！

黃敬群 (jserv)

黃敬群（網路暱稱為“jserv”，現為新酷音輸入法計畫維護者，並從事 GNU GCC 與若干自由軟體開發）<http://jserv.sayya.org/>

進入自由軟體世界的大門

這個世界真是奇妙，您永遠不會知道您現在所做的事情會對他人產生多大的影響。就如同飛舞中的蝴蝶，不知道它每分鐘拍動翅膀累積在大氣的力量，竟會造成一年後北美境內的龍捲風一樣。像這樣的蝴蝶效應，每分每秒都在對這個世界產生微妙或是重大的影響，就如同 Richard Stallman 的 GPL 及 GCC，亦或是 Linus Torvalds 的 Linux kernel，一個當下的善意，竟會對幾年後的這個世界產生如此重大的革命。

從事 GNU Debian Linux 以及 Open Source 推廣運動及教育多年，也因此結交了 KaLUG TnLUG 及許多 FS/OSS 志同道合的夥伴，萬萬也想不到幾年前的善意，而結識了聖璋，竟也間接影響了一本書的誕生。一本書的誕生，背後所投注的心血是十分巨大的，而一本心力所鑄，躺在書架上的書，它可以發揮的影響力更是比一個活動或是一個教育學程來的更大。而它將靜靜的躺在世界某個角落散發它的影響力，等待您的發掘。

想學好 Linux，首先要先成為它的信徒，一件印有 FS/OSS 圖騰的衣服是必要裝備，擁有一台裝有 Linux 的 NoteBook，並在上面裝好華麗酷炫的 Xwindow，時時刻刻帶著它使用它，讓這台 NoteBook 成為您生活必需品，並隨時隨地向您的朋友洗腦以及炫耀您的電腦桌面有多麼的酷炫漂亮。如果您能做到這樣的地步，那我要恭喜您已踏入了 Linux 的大門，之後我會建議您就應該開始懷疑與質疑它，並面對及承認它的缺點，進而改變它。它最大的優點不在於它所向無敵以及完美毫無缺點，相對的它的缺點還挺多的，它有個人人都有權力改進及持續進化的優點，只要您想您願意，您有能力，您就可以取得及窺視它的奧秘，FS/OSS 最大的優勢在於它多軌平行競爭，而且是正向循環進化。

常有人問我推薦一本學習 Linux 的好書，我一貫的答案都是，去書局翻一本您看得懂的書，重點不在於好壞，厚薄，或是有沒有深度。重點在於您是不是看得懂，看得懂您才會看，會看才會看得完，看得完對您才是好書。看完後請放這本書自由，火車站咖啡館都是將書本放生不錯的地點，然後再去找下一本您看得懂的書。

衷心希望這本書對於正在閱讀的您會有所幫助，也希望這書可以對您產生正向效應，可以引領您進入自由軟體世界的大門成為它的信徒。

黃宇新 (lloyd) 於高雄河堤

Welcome to the Most Friendly Communitty in the World, Ubuntu@Taiwan !

當初認識這本書主要作者之一 dbtsai 是因為經營 Ubuntu@Taiwan 社群及系列社群支援站台的關係。在某天收到了這位（原本只看 ID 覺得應該滿嚴肅的）仁兄的信表示有意願加入幫忙，卻在回信後就消失無蹤兩個月，正感洩氣之時沒想到收到了本書的另一位主要作者 DUCATI 的聯絡，表示與 dbtsai 兩人已經開始交往。噢不是，是已經開始著手寫作一本專為獻給中文世界的 Ubuntu User

的入門書，我又驚又喜，約了幾次的聚會熟識後才發現人不可 ID 相，原來這兩位其實都是很可愛的有為年輕人。見到人才，想當然爾就明目張膽地設計讓他們與 DocTeam（文件組, Ubuntu@Taiwan Documentation Team）合作，結合，然後說服他們把他們吃掉納為我們的成員之一！（好吧，我承認聽起來我很壞。但為了讓更多麻瓜變成巫師^❶，我不入地獄誰入地獄呢？）

現在時髦的 DUCATI 是 DocTeam 和 ActTeam（活動組）的活躍成員，不但是我這清純小生在社交的課題上的良師益友，也是很令人期待的文件創作者。而高大卻害羞可愛的 dbtsai 同時是 DocTeam 和 PkgTeam 的成員，雖然說他最近因為研究所的課業壓力而能待在 IRC 線上聊天的時間變少了，煞是可惜。

❶ 在有些電腦人團體中，會以「麻瓜」一詞來形容對電腦不慎瞭解的一般使用者（End Users），並以「巫師」一詞來形容能用電腦如使魔法般創作出嶄新事物或對電腦內部極為瞭解的人（當然，通常是某種形式的技客（Geek））。

我不得不說，這應該是我看過目前市面上最好的一本 Ubuntu 的原生中文書，不像其他有些電腦書量產團體，為了趕出版時間和著作量搶市場而造成品質的粗略，這本書有深入淺出的技術細節描述，又有著平易近人的文字運用，讓初學者能學建置一個舒服無負擔的環境。

另外 Canonical 官方也有出一本英文的 Official UbuntuBook，有能力讀英文書的朋友們也可以一起買來參考看看，當然，若您願意，我們也非常歡迎您加入我們的大家庭，文件組（DocTeam）和軟體特包隊（PackageTeam）和其他幾個小組都很缺熱情的新血呢:D

希望大家能由此作者們的心血結晶開始，一步步地發現瞭解 Ubuntu Linux 這個世外桃源並重獲新生。

最後，還是要說我們的歡迎詞：

Welcome to the Most Friendly Community in the World, Ubuntu@Taiwan !

Best Wishes,

Ubuntu@Taiwan Founder/Leader, BlueT.

--

/ 0x5296C7A8 /

/ matthew = BlueT /

/ matthew@BlueT.org /

/ Just be Perl Hacking! / (IRCnet #Kaohsiung.pm)

作者序

李聖璋

說起 Linux，為什麼會想用 Linux 當成平常用的桌面環境呢？這就要從幾年前說起了！當時的我剛上大學有一陣子了，在學校的課也覺得無趣，而且有個單字出現在我的腦海裡很久了。嗯，那就是 Linux，當時不認識 Linux 的我，印象中只知道 Linux 好像是用來架站用的？但！Linux 倒底是什麼呢？好奇心強的我，也喜歡跟別人用不一樣的東西，當下就下了一個決定！不管 Linux 是什麼東西，一定要去學學看！在老天巧妙的安排，小弟我當了 Lloyd 老師的學生，而且老師教的是 Debian，真是讚啊！在老師細心教導讓我和 yenjinc 們有了很好的基礎，得以進一步的學習。數年後今天小弟覺得，我們從老師身上學到最多的是 **OpenSource 的精神**！讓我們受益不少！雖然 Linux 自己看書就可以學會的，不過套句 Lloyd 老師說過的一句話「學 Linux 就是要多交會 Linux 的朋友才好玩」。

那時候我們幾個有 Note Book 的同學，有時候也會拿著自己的 Note Book 出來互相炫耀。互相討論在 Linux 上使用的各種問題，說真的去上 Lloyd 老師的課比上學校的課還開心多了 ^ ^，後來也努力的考了 LPIC。但這並不能阻止我對 Linux 的渴望。於是小弟我去上了 level 2 的課程，又在老天的巧妙安排下，剛好去上了 Netman 老師的課，其實是小弟我事先打聽過的，然後也進了 TnLUG 當苦力小弟，不過！老實說並不苦啦，當時只是去佈置場地，幫忙找講師。而且 LMAN 長輩還會請吃飯哦！所以想要對社群有所貢獻的朋友們，趕快加入 TnLUG，就會有大哥請吃飯呢！不過當然要有所付出哦。

接著在一場 PCMan 到 TnLUG 演講的場次，認識了 dbtsai，兩人相約拿著 Note Book 去咖啡廳聊天，在 dbtsai 的勸說之下決定試用 Ubuntu，言下之意也決定來寫 Ubuntu 的書。在這裡真的要特別感謝 dbtsai，如果不是 dbtsai 起了頭，就不會有這本書的出世了。而這本書的完成也不是只靠我們兩人的力量，在這裡我們要感謝 Tim 先生，給我們這次出版的機會。

致謝

感謝幫忙趕稿的 Haway 與企鵝狂，還有 yenjinc 的稿子，lancetw 也提供了一些自己使用上的意見，jserv 大長輩更是辛苦！連夜的幫忙校稿犧牲睡眠的時間，PCMan 大哥也是，也幫忙測試書上的步驟。還有 BlueT，好友兼 Ubuntu TAIWAN 站長，讓我們有機會對社群有所貢獻。當然囉！不是只有這些朋友，我想這個可能要列個表才寫得完吧。不過最重要的當然是媽媽，辛苦的工作真的很心疼，我只能說現在車不好賣啊！

在這裡我也要特別感謝我的好友榮玲，有了妳的鼓勵才可以讓我堅持做完每天的工作 ^_^。

本書獻給我的母親與在天上父親

李聖瑋 ducati lee 謹致
ducati@ubuntu.org.tw
2006/11/20 於 彰化

作者序

蔡東邦

在一年多前陽光燦爛的台南夏日，我開始構思寫一本有關 Debian 的書，當時市面上大部份的 GNU/Linux 書籍都是介紹 RedHat 或者 Mandrake 這類使用 RPM 套件管理系統的 Linux 發行版。所以當時我找幾個在成大實驗室的學長和學弟打算在暑假合寫一本 Debian 書籍，但是年少輕狂，總覺得寫寫書是很容易的事，當然這個計劃就僅止於打嘴炮的階段。

接下來準備研究所推甄，更是把這個計劃忘記了！直到推甄上後，在某一場研討會認識了 ducati，我們常常帶著 Notebook 一起去咖啡廳討論最新的技術，等熟了一點，我就試探性的問他是否有興趣合寫一本有關於 Debian 的書籍？他也鑑於市面上 Debian 體系的書籍太少，就被我說服了。那一陣子，我開始使用 Apple ibook，但是當時 Debian 要裝上去有點麻煩，很多細節需要自己手動設定，老是讓記性不好的我弄很久。但是筆者又很喜歡 apt-get 系統，裝任何軟體只要一行指令，就可以自動解決 Linux 相依性問題，找來找去，我就愛上 Ubuntu 了！

Ubuntu 有著和 Debian 一樣讓人稱道的套件管理系統，在安裝過程中又免除 Debian 需要凡事自己來的瑣碎，所以那時候立刻改成寫 Ubuntu 的書籍。現在想想，當時真的是壓對寶了，尤其是目前 Ubuntu 已經變成了全世界最流行的發行版之一，筆者看著它長大，特別有感觸。

接下來的半年，筆者和 ducati 開始構思書本要如何呈獻給讀者，如何把艱澀的概念告訴讀者，而不是僅僅告訴讀者指令就是該這樣打，沒有為什麼！尤其筆者

是學物理的，特別討厭沒有道理的事務，因此本書在我們兩位作者的構思下，將會用故事和範例鋪陳整個架構，讓讀者可以知道為什麼要這樣做而不是只知道這樣做就可以 work。筆者認為這樣的寫作方式可以建立讀者正確的 GNU/Linux 學習觀念，之後遇到問題可以觸類旁通。

筆者在章節上的規劃是先介紹 Ubuntu 的始祖 Debian，接下來介紹各種不同的 Ubuntu 分支和 Ubuntu 創立理念，讓讀者對於 Ubuntu 有充份的歷史背景知識。

接下來介紹 Ubuntu 的安裝方式，基本網路設定，和 Ubuntu 最 cool 方便的套件管理系統。在書中，也有目前最流行的 3D 桌面安裝方式，果凍般的視窗效果和可以旋轉的虛擬桌面，以及大部份 Linux 書籍很少提到的桌面應用，甚至是使用 Linux 看有字幕的 avi 影片，這大概是真正的 Linux 社群玩家才會比較常接觸到的。在書中我們探討了在 Linux 底下使用 wine 模擬玩 Windows 的《魔獸世界》等遊戲，這也是本書和別本書比較不一樣的地方。

當然 GNU/Linux 的強項是在伺服器應用，所以本書也不例外介紹了 Apache，NFS、FTP、samba 和 DHCP 等常見的應用。尤其是 Apache 筆者引入的網頁壓縮的技巧，配合 php 加速程式，在筆者的測試可以讓您的網頁快上個好幾倍，同時上線人數大約可以多個二到三倍左右，依您的網頁型態而定。

在這麼多的內容中，筆者也在寫作中學習不少，尤其是很多前輩的指教，更讓我獲益匪淺。如原本筆者在成大時的學長黃天祥，本來也是本書作者之一，無奈因為成大資管博士班課業繁忙，最後無法參與本書撰寫。但是他在本書的寫作風格和校閱方面，幫上了許多的忙。

我的好同學兼好朋友黃詩姍，若是沒有她細心的幫我訂正錯字，那麼這本書將會成為貽笑大方的典型錯字標準範例書。

也要感謝我的好朋友 ducati，尤其是接近截稿日的時候，因為忙於學校考不完的期中考和改不完的作業，我幾乎沒有幫上什麼忙！若是沒有他的話，這本書可能會無疾而終。

感謝各位 GNU/Linux 界的前輩，看得起我們所著的書，願意耗費許多心力來幫我們校稿和修正錯誤，並且願意幫我們寫推薦序。尤其是 jserv，PCMan，鳥哥這幾個社群中鼎鼎有名的高手，耗費許許多多的夜晚和筆者們討論和除錯本書的內容。

BlueT－Ubuntu TW 的站長，也提供了許多的資源讓我們使用，尤其是我們這幾個人常常一起喝個小酒，聊聊未來 Ubuntu 的發展，更是堅定了彼此的友誼。

謝謝台大物理所 307 碩一自習室所有陪伴我度過苦悶研究生日子的朋友，讓我在苦悶的時候還可以很開心的和大家討論功課，解解悶。

感謝上奇出版社 Tim 莊吳行世先生，讓我們有機會可以把我們的心得出版成一本書，並且在寫作的過程中，不吝於給與指導，並且常常幫我們解決排版和寫作問題。

本書獻給辛苦養育我成人的父親與母親

蔡東邦 dbtsai@gmail.com

2006/11/20 台北

CHAPTER

1

Ubuntu GNU/Linux

Ubuntu 這個 Linux 發行版，在最近紅透半邊天了，越來越多人使用它來取代日常的工作和企業伺服器的使用。那麼到底它有什麼魔力呢？它的特點就是簡單易用，其中最重要的部份是繼承了 Debian 眾多套件的資源，也就是大部份 Debian 裡面有的套件，都可以在 Ubuntu 裡面找到。並且 Ubuntu 有著和 Debian 一樣的優秀的套件管理系統，初學者可以擺脫一般發行版安裝軟體的相依性惡夢和 Debian 裡面樣樣設定都要自己來的麻煩。也因此 Ubuntu 在初學者中漸漸紅了起來，許多高手為了省掉安裝設定的麻煩，也開始使用 Ubuntu 了。

備註 目前從 DistroWatch.com 這個網站統計了目前世界上有多少 GNU/Linux distribution，並且也對各個 distribution 做出熱門度的排名，在目前的排名中，Ubuntu 是拔得頭籌。資料網址為 <http://distrowatch.com>

接下來我們將先對本書主角 Ubuntu 的祖先 Debian 作一些介紹，畢竟 Ubuntu 很多使用的特性和優點都是繼承於 Debian。最後筆者將會告訴你們，為什麼目前國外最熱門的是 Ubuntu，與國內正在開始流行的 Ubuntu 正是你們最佳的選擇。

1.1 開山祖師－Debian



圖 1.1 Debian Logo

話說 1993 年 8 月 16 號，Debian 計劃在 Ian Murdock 的帶領下，正式開始運作。而大家或許會好奇，Debian 這個名字是怎樣來的。這是因為 Ian Murdock 的老婆叫作 Debra，而他是個非常疼老婆的人，所以就取他老婆名字的前 3 個字母，Deb，再從他自己的名字取 3 個字母，Ian，合起來就是 Debian 啦！

Debian 是一個完全由社群所維護的 Linux distribution，它背後不像其他的發行版有營利組織給與協助。因而可以說是最乾淨和精簡的 GNU/Linux 版本。而 Debian 最為人所稱道的是優秀的套件管理系統，和整合了眾多的應用軟體，也使得很多 Linux distribution 是擷取它的優點分支出去。但是被很多人詬病的問題是，Debian 的學習曲線比較高，使用難度也相對較高，造成了入門的門檻偏高，所以被視為較進階的 Linux distribution。

若談到 Debian 的命名方式，則是很有趣的一件事，如表 1.1。所有的研發代號都是使用 Pixar 公司的動畫電影《玩具總動員》裡面的主角來命名。筆者很好奇，當所有角色用完時會怎麼辦呢？而版本就是正式釋出時所使用的代號，不過有時候大家會把開發代號和版本混在一起講。而為什麼 Sid 是永遠的未來呢？那是因為在電影中，Sid 是玩具終結者，所以大概是用它來象徵還未穩定 (unstable) 版本的開發代號吧！基本上 Ubuntu 是以 Sid 來作其基礎，並且加以穩定化，然後再加上一些獨有的套件，所以 Ubuntu 裡面的程式是頗新的哦！

表 1.1 Debian 發展年表

版本	開發代號	釋出時間	詳細描述
1.1	Buzz Lightyear	1996 年 06 月 17 日	巴斯光年是穿太空服的玩具
1.2	Rex	1996 年 12 月 12 日	電影裡的暴龍
1.3	Bo Peep	1997 年 06 月 02 日	電影中牧羊女芭比
2.0	Hamm	1998 年 07 月 24 日	影片中的小豬撲滿
2.1	Slink Dog	1999 年 03 月 09 日	影片中的玩具狗
2.2	Potato	2000 年 08 月 15 日	影片中的 Potato Head 先生
3.0	Woody	2002 年 07 月 19 日	玩具警長胡迪，是電影主角之一
3.1	Sarge	2005 年 06 月 06 日	綠色塑膠玩具士兵的首領
3.2(預定)	Etch-a-Sketch	下一個版本，時間未定	影片中的玩具黑板
無	Sid	永遠的未來	電影中的玩具終結者

經過十多年的發展，Debian 團隊加入了許多有用的管理程式，例如最有名的 APT 系統，還有很多獨樹一幟別無分號的工具程式。而它最特別的地方是它和其他發行版包山包海的情況不同，而是有點像玩樂高積木一樣，要自己慢慢架構上去，這是與 GNU 一貫自由的手法與訴求相仿，可讓使用者選擇自己所想要裝的，而不去強迫使用者一定要安裝，這也是為什麼 Debian 安裝起來比較簡潔乾淨的原因。但是這樣堆堆樂的概念，也是讓很多新手卻步的原因。但是在這十幾年來 Debian 不變的精神是自由與使用的穩定性，這就是為何國外很多商業公司和政府組織等使用 Debian 來當作伺服器等應用。

備註 使用 Debian 的機關名單，可參見以下網址：

<http://www.debian.org/users/>

或許目前使用 Debian 的人數還沒有其他發行版本多，但 Debian 的愛好者大部份都是有經驗的玩家，因此筆者認為 Debian 在社群資源上是沒有問題的，唯一的問題是學習曲線比較高。

這時候對初學者怎麼辦？答案當然就是我們接下來要談的 Ubuntu 啦！因為 Ubuntu 有著和 Debian 一樣的優秀血統，又那麼容易安裝，對新手老手都合適。因此就有人問幹嘛還要有 Debian 呢？統一大家都用 Ubuntu 就好啦！不過筆者認為，畢竟 Ubuntu 是從 Debian 分支而來的，所以 Debian 一定有自己的特色和愛好的人。的確是如此，高手練劍，首在練氣。所以要升到高手級時，要先把外功（Ubuntu）練好，再來練內功（Debian），不然內功未成，就一直挨打，沒半點外功，就容易被打死（放棄 GNU/Linux）。所以筆者希望大家都可以如此到達內外俱修的太極張三丰境界喲！

1.2 承先啟後，Ubuntu

1.2.1 Ubuntu 的創始



圖 1.2 Mark Shuttleworth 在國際太空站



圖 1.3 Ubuntu 的理念，同源一體

Ubuntu 的創始人 Mark Shuttleworth 於 1973 年 9 月 18 日在南非出生，他擁有 Cape Town 大學財務和計算機系統學位。Mark Shuttleworth 本人是程式設計師，也是 90 年代 Debian 早期的開發者。早期他設計一些網路安全程式，並且在 1995 年成立了 Thawte 公司。這間公司主要營利是來自於數位認證和網路安全等業務，Mark Shuttleworth 在 1999 年 12 月把公司賣給了 VeriSign，獲得了約 5.6 億美金。隨後他又成立了 HBD Venture Capital 公司，從事一些風險投資和產業培養的事業。在 2001 年，他成立了 Shuttleworth 基金會，這個非營利性組織主要是推動南非教育和 Open Source 的推廣。傳奇的是，他在 2002 年 4 月 25 日，自費花了兩千萬美金，搭乘俄羅斯 Soyuz TM-34 太空船，來到了國際太空站，成為非洲人中第一個上太空的。接著 Mark Shuttleworth 於 2004 年創立了 Canonical Ltd 公司。這個公司贊助成立了 Ubuntu GNU/Linux 這個發行版。而 Mark Shuttleworth 現在認為 Ubuntu 是他畢生的志業，為了能把所有時間都投入 Ubuntu 的研發，他甚至把他所有財產交給專業經理人管理，讓他不用擔心理財等雜務。他於 2005 年又以一千萬美金成立 Ubuntu 基金會作為運作資金，讓這個計畫可以在萬一 Canonical Ltd 公司無法繼續支援時仍舊可以順利運作。

那麼 Ubuntu 這個字到底怎樣來的呢？其實這是一個很古老的非洲話，代表著對人仁慈，因為我們是同源一體的！所以 Ubuntu GNU/Linux 也就是本著這樣的信念，來當成此發行版的研發綱領。因此 Mark Shuttleworth 保證說不論企業版

還是個人版，都永遠不會收費。所以 Canonical Ltd^❶ 公司將以諮詢和企業服務來作為營收。

Ubuntu 目前固定每年 4 月份和 10 月份發行新版，而且新版的發行版都同步包含當時最新的 GNOME 視窗系統。它第一個發行版發行日期是 2004 年 10 月 20 日，研發代號是 Warty，版本號是 4.10。有趣的是，它每個版本號碼組成的整數部份是發行的年份，小數部份是發行的月份。但是應該在 2006 年 4 發行的 6.04 版，為了改善多國語系支援，於是延遲到 6 月，變成了 6.06 版，也因此打破了慣例。

表 1.2 Ubuntu 發展年表

版本	開發代號	釋出時間
4.10	warty (Warty Warthog)	2004 年 10 月 20 日
5.04	hoary (Hoary Hedgehog)	2005 年 04 月 08 日
5.10	breezy (Breezy Badger)	2005 年 10 月 13 日
6.06	dapper (Dapper Drake)	2006 年 06 月 1 日
6.10	edgy (Edgy Eft)	2006 年 10 月 26 日
無	grumpy (Grumpy Groundhog)	永遠的未來

因為 Ubuntu 更新的很快，每半年出一次新版，所以大部分的人會使用套件管理系統來更新系統。方法是直接修改套件管理系統內使用版本的定義，例如你原來是使用 5.10 (Breezy Badger)，現在出了新的穩定版，6.06 (Dapper Drake)，那你只要把套件管理系統內的 `sources.list` 檔案裡面所有的 `breezy` 字眼改成 `dapper`，然後再執行下列指令：

```
$ sudo apt-get update && apt-get dist-upgrade
```

就可以享用你的新系統啦！在設定檔 `sources.list` 裡面，版本設定是用開發代號的第一個單字，例如你使用 6.06，那麼在 `sources.list` 應該是定義成 `dapper`，並且全部小寫，詳細方法筆者會在後面套件管理系統的地方說明。

❶ <http://www.canonical.com/>

通常在 Ubuntu 內，使用者都是使用穩定版，如果你高興的話，可以每半年用上面的方式無痛升級。在 Ubuntu 中，使用不穩定的版本對一般使用者而言，是不建議的。因為 Ubuntu 出得很快，所以不會如 Debian 每個穩定版過度嚴謹、數次延遲推出新版本，以至於許多使用者得改用不穩定的版本。因此，無可避免地，Ubuntu 每個穩定釋出版本間或多或少會存在一些瑕疵，但可透過系統升級修正。當然你也不一定需要每半年更新一次整個主機，從 Dapper Drake 開始，Ubuntu 提供了桌面版 3 年和伺服器版 5 年的安全更新服務。所以就算你沒有更新到最新版，安全性的問題，Ubuntu 一樣都會幫你解決喔！

1.2.2 Ubuntu 組件庫的哲學

因為 Ubuntu 衍生於 Debian，所以也是對於自由很堅持，因此希望把自由軟體和非自由軟體拆開來放。有些是因為開發資源考量，有些套件是在 Debian 裡面有，但是不被 Ubuntu 官方支援的。所以 Ubuntu 的處理方式是把它們拆成以下四大主要組件庫，類似 Debian 一樣，讓使用者選擇需要。

- **main**：大部份常用的自由軟體都會收編在 main 裡面，這裡面所含的軟體需要完全符合 Ubuntu 對自由軟體的定義，詳細規定請見 <http://www.ubuntu.com/ubuntu/licensing>。因此通常使用 main 就可以應付日常所需。而 main 裡面所有軟體，都是官方支援的，所以 Ubuntu 將會提供 main 的安全更新和技術支援。而它的角色和 Debian 裡面的 main 差不多。
- **restricted**：裡面所含的是非自由軟體，通常是不公開程式碼的程式或者授權不符合 Ubuntu 的要求。例如顯示卡驅動程式等等。雖然是非自由軟體，但是因為這些都是比較重要的非自由軟體，所以 Ubuntu 在符合該軟體授權下，將會進行穩定性上的測試，並也提供安全更新。因為不是自由軟體，也無法對原始程式碼作深入修正，所以 Ubuntu 在遇到問題時不一定能解決，所以維護品質沒有 main 那麼好。在桌面應用中，顯示卡加速驅動程式或者 Flash 外掛等都是包在這一類套件庫內，因此桌面使用者通常都會使用 Restricted 這個組件庫來增加多媒體支援的能力。
- **universe**：這裡面大部分的軟體都是自由軟體，而且幾乎都是包含在 Debian 裡面的 main 中。但是因為人力上的問題，所以 Ubuntu 無法親自去包這些套件，而是直接由 Debian 裡面抓下來重新編譯，或者是直接從

外部程式碼自動編譯出來的。也因此，Ubuntu 無法對這樣的軟體套件提供正式支援，Ubuntu 雖然有提供它的安全更新，但是不保證能即時修正或能穩定運行。例如國內知名的 BBS 連線軟體 PCManX 就放在這裡面，或國內有名的輸入法程式 gcin 也是包含在這裡面。Ubuntu 預設沒有啟用 universe 套件庫，所以想要用的人得自己在 sources.list 啟用。

- **multiverse**：裡面的軟體都是非自由軟體，但是因為重要性不足或授權的因素，所以放在這裡。而這裡的軟體也是不被官方正式支援的。而和 universe 一樣，Ubuntu 預設也是沒有啟用它。

前兩個 Main 和 Restricted 是官方正式支援的，Universe 和 Multiverse 是不被官方正式支援的。而 Restricted 和 Multiverse 裡面是 non-free 的軟體。除了以上這幾個組件庫以外，它還有其他附屬的組件庫。因為這些組件庫在穩定版出來以後，就不會再更動，所以當有漏洞的時候怎麼辦？或者有人懶得每半年都更新一次，但是又想要在舊版 Ubuntu 裡面享用一些比較新版的軟體時，總不能自己編譯和搞定 library 的問題吧！所以在上面四大組件庫，都有以下三個分支。

- **security**：穩定版本正式釋出後，如果程式有重大安全性上的問題，將會在這裡更新。這裡所謂的更新，通常是指提供對原始程式碼的修正並重新編譯過。還是同一版本，所以功能也不會增加。因為 security 套件庫是針對系統安全而更新，所以裡面大部份更新都和伺服器或系統架構有關。而這個組件庫預設是被開啟的。
- **updates**：這也是在穩定版釋出後，提供系統更新的一個方式。但是和 security 不同的是，這裡所做的更新都是和安全性沒有關，而是和程式的 bug 相關。例如某套桌面軟體有時候會當掉，而這個當然和系統安全沒關係了，這種更新就會放在這裡。這裡所謂的更新和 security 一樣，只修正原始碼，並不會增加新功能。這個組件庫也是預設被開啟的。
- **backports**：例如 6.06 (Dapper) 中附的網頁瀏覽器是 Firefox 1.5.0.3，但是在穩定版釋出後，也許這段時間裡面出現了 1.5.1、1.5.2、1.5.3 等版本，但是在主套件庫中軟體是一定不會更新的，而預設開啟的 security 或 updates 套件庫除非是有安全性的問題或者是有些 bug，否則不會更新。就算是因為安全性或 bug 而更新，也通常不會換成新版的。所以 backports 就是為了解決這樣的問題，開發者會將新版的軟體就會放在 backports 中，讓使用者使用舊版發行版但是可以用到比較新的套件。而通常在

backports 中的程式都是和桌面相關，而不會去更新到系統或伺服器架構，這樣才能保證系統運作穩定。而預設 backports 是沒有使用的，所以要自己修改 sources.list 來啟用。

備註 使用者可以要求希望哪些軟體進入 backports，詳細辦法可以參閱 <http://backports.ubuntuforums.org>

1.2.3 各種不同的 Ubuntu 分支

Ubuntu 原先釋出的版本是預設採用 Gnome 這套桌面環境，但是在 GNU/Linux 的世界裡面有許多種的桌面系統，或者因為版本的訴求與某些使用理念不同，所以 Ubuntu 目前有五種不同的安裝版本可以使用。而這幾種安裝版本的組件庫是採用一樣的，它們可以採用不同的光碟來安裝，安裝後就是預設為不同的衍生版本，因為採用同樣的組件庫，所以你當然可以在安裝後切換成別的衍生版本。實際上各個衍生版本都是以一個叫作虛擬套件的套件包來控制。也就是那個虛擬套件是空的，但是因為裡面設定了相依到很多該衍生版所需的套件，所以安裝那個虛擬套件，就會自動把該衍生版所有套件裝起來。安裝方式就是使用套件管理程式，就可輕鬆的安裝好了唷！當然這是給各位一個概念，詳細的安裝說明會在後面提到。

- **Ubuntu 桌面系統**：這是最主要的衍生套件，使用 Gnome 桌面為基礎，也是歷史最悠久的，使用的人也是最多的。本書整個主題大致上都是以它為中心，所以建議讀者先玩玩看這個衍生套件吧！而它的相依虛擬套件就是 ubuntu-desktop，所以如果你要從其他的環境轉成使用 Gnome，就只要下以下指令，系統就會幫你弄好相關設定和安裝啦！

```
sudo apt-get install ubuntu-desktop
```



圖 1.4

- **Kubuntu 桌面系統**：這個衍生版大概是最熱門的，它使用 KDE（喜歡使用 KDE 的人以歐洲人比較多，因為 KDE 就是原生於歐洲。而 Gnome 的開發者比較多在美洲，所以美洲人就比較喜歡用 Gnome 了）。來當成桌面環境。所以 Kubuntu 這個衍生版本是為了讓那些不喜歡 Gnome 的使用者可以有裝好就是 KDE 的使用環境所發行的。而它相依的虛擬套件為 kubuntu-desktop，所以要從其他環境裝 Kubuntu 只要下以下指令，就會一切幫你搞定了。

```
sudo apt-get install kubuntu-desktop
```



圖 1.5

- **Edubuntu 桌面系統**：Edubuntu 是 Ubuntu Linux 的教育分支版。它主要的訴求是教育使用，所以裡面有很多精心挑選的教育相關套件，和一些小遊戲等。家裡面有小孩子的話，可以抓下來玩一玩。而為了可以讓教師們快速安裝到電腦教室或安裝到老舊的電腦裡，它還使用了 LTSP 技術，可以讓老師們在短短的一個小時內建立整個電腦教師環境。會有這個分支的原因是 Mark Shuttleworth 始終對教育有著熱誠，所以他希望將 Ubuntu 引進到南非弱勢的小孩身上，而使得他們能學習和改變現狀。相依的套件是 edubuntu-desktop，所以要從其他環境安裝 Edubuntu 只要打以下指令，就可以了。

```
sudo apt-get install edubuntu-desktop
```



圖 1.6

備註 LTSP (Linux Terminal Server Project) 的發展目的是提供多個工作站只負責顯示，實際軟體和計算在伺服器上面，也叫做 thin-clients 機制。它的好處是工作站可以使用很老舊的電腦，因為工作站只用來當 XTerminal，所以一切資源都放在伺服器上。這個方式可以讓維修成本和管理成本減少很多，所以特別適合公司和學校使用。

- **Xubuntu 桌面系統**：因為它採用了比較輕量級的桌面環境 Xfce，所以這個版本的訴求是針對比較老舊的電腦。Xfce 的設計理念是減少使用的系統資源，並且成為一套完整可用的視窗管理程式。而它使用了 gtk 2.x 的函式庫，所以它對於 Gnome 上的程式相容性很好，甚至連 Xfce 的佈景主題也可適用於 Gnome 應用程式。相依的套件是 xubuntu-desktop，所以要從其他環境裝 Xubuntu 只要打以下指令，就完成啦！

```
sudo apt-get install xubuntu-desktop
```



圖 1.7

- **Server 專用版本**：適合在伺服器上面使用。使用安裝光碟開機後，甚至有一個選項可以快速配置好 Apache + PHP + MySQL，所以對於需要很快的時間內裝好伺服器的人，它是一個不錯的選擇。筆者已經拿它來裝在學校的幾個大型伺服器上，認為它的穩定性很好。例如筆者安裝在 IBM eServer xSeries 206 伺服器只花了我約 15 分鐘，且它的設定配置也是很簡單。預設安裝完後只有文字模式，所以如果要拿 Ubuntu Server 來當成伺服器要有點基礎，至少要知道設定檔怎樣設定。當然要裝上桌面系統也是可以的，看你想要使用 Gnome 或 KDE 等，相信對讀者而言應該不難。

1.2.4 Ubuntu 支援的不同硬體架構

GNU/Linux 因為開放原始碼，所以我們可以很容易在不同平台上編譯來跨平台。目前 Ubuntu 已經支援了四種平台，讀者可以視需要下載不同的平台版本。記得喔，拿錯平台的光碟是不能安裝開機的喔！

- **PC (Intel x86)** : 目前讀者外面買得到的電腦幾乎是相容於這個平台，包含 Intel 系列和 AMD 全系列都可以安裝。建議新手或者不知道我在討論什麼的人直接使用這個版本，保證一定可以跑！
- **64-bit PC (AMD64)** : 不要看到 AMD64 就以為只有 AMD 64 位元的電腦能用唷！因為這個指令集是 AMD 先發明的，所以才用 AMD64 來當名稱。目前支援這種指令集的機器有 AMD K8、AMD Opteron 全系列和 Intel EM64T 的 CPU。若在桌面上任務使用 64 位元作業系統的話，有些沒有開放格式的视频將不能播放，並且也沒有 Flash plugin。但這些電腦都相容於 32 位元模式，所以筆者建議降級安裝 32 位元的作業系統，你會遇到較少的麻煩。當然有人是採用在 64 位元作業平台上建立一個 32 位元子系統，然後再用 chroot 的方式去執行這些軟體，這樣可以保有兩樣優勢，不過這不在我們討論的範圍內。
- **Mac (PowerPC)** : Apple 使用 PowerPC CPU 的電腦，都需要安裝這個版本。例如 G3、G4 和 G5 系列。
- **Sun UltraSPARC** : 這是很少見的機器，大部份只用在實驗室或一些企業工作站使用。

1.2.5 Ubuntu 體貼的兩種安裝方式

很多人在推廣 GNU/Linux 的時候，都會遇到一個問題，假如要讓人家看 GNU/Linux 的效果時，都要安裝，那不就太麻煩了嘛！或者誰知道 Ubuntu 合不合我用阿！如果想要嘗試 Kubuntu 或 Edubuntu 都要在安裝，那不就太累了嘛！搞不好安裝了還會搞壞原來的系統，所以 Ubuntu 體貼的把 Live CD 和安裝光碟整合在一起。所以用 Ubuntu 的安裝光碟開機後，它就直接進到 Xwindow，不需要經過安裝，就可以嘗試看看 Linux 的威力唷！這個概念有點類似展示片，當你試用覺得很好用以後，只要點桌面上安裝的圖示，它就可以安裝到你的硬碟上啦！

而如果你不喜歡進到 Xwindow Live CD 以後，再按桌面上安裝的圖示來安裝，或者第一種方式安裝失敗，Ubuntu 也有提供另外一種傳統模式的選擇，也就是純文字介面的安裝方式。筆者還是喜歡採用文字介面來安裝，這種安裝方式速度比較快，比較不耗系統資源，也是最傳統的方式唷！

這裡做一個總結，避免讀者迷失在上述的專有名詞或術語中。講了那麼多，有 Ubuntu、Kubuntu 和 Xubuntu 等分支，除了這些又討論到不同平台，而分別又有 AMD 64、x86 和 PowerPC 等，每種都分別有 Install + Live CD 整合光碟和文字模式安裝光碟。所以算一算很多種光碟 iso 燒錄檔可以下載，天曉得該抓那一張光碟阿！當然看完前面有概念的讀者，可以選擇自己的喜好來抓。而對於新手我建議先玩 x86 版本的正統 Ubuntu，也就是底下兩個啦！

- **ubuntu-y.mm-desktop-i386.iso** : 這個是使用 Gnome 當桌面的 Live CD + 安裝整合光碟，放進去開機可以在你安裝前，先試用看看。喜歡的話，點桌面上的安裝圖示就可以安裝啦！
- **ubuntu-y.mm-alternate-i386.iso** : 這是傳統的文字模式介面安裝光碟，安裝後基本上和上面那一片一模一樣。雖然這片沒有漂亮的安裝介面，但是安裝速度較快，對於硬體相容性也比較好。如果用上面那張有問題，可以嘗試看看這一片唷！

備註 y.mm 是版本號，例如在 edgy 中就是 6.10，也就是 y 是釋出的年，mm 釋出的月份。所以這個例子的燒錄印像檔的檔名就是 ubuntu-6.10-desktop-i386.iso 和 ubuntu-6.10-alternate-i386.iso

而取得官方原廠光碟或者從網路上下載映像檔並且燒錄的方法，我會在後面慢慢地告訴你們，精彩的在後頭哩！

1.3 GNU/Linux 學習資源

1.3.1 解決問題的心態與方法

雖然說 GNU/Linux 是開放原始碼，所以遇到任何問題，理論上都可以去 hack 或研究原始碼，就可以找出問題所在。但不是每個人都是程式設計師，也不是都有閒情逸致去研究程式碼吧！而且大部分的問題，不是程式上的問題，而是使用者操作或設定上有問題，才導致系統有錯。所以在 GNU/Linux 上，遇到問題的話，有兩種，一種是程式碼上面本身就有錯誤，另外一種是使用者自己不會用的問題，而大部份使用者很少會去碰到原始程式就有問題的情形。所以不要千錯萬

錯都是 GNU/Linux 錯，整天用抱怨的其實是沒有什麼建設性的，大部分的 bug 是可以解決的。因為剛開始學習 GNU/Linux 時，這是你剛接觸的系統，畢竟大部分的人都使用 MS Windows 有一大段時間了吧，也都多少有些被制約了，想說某個功能就是這樣，但是這在別的作業系統可能是要用別的方法。

記不記得當年你開始學習 Windows 也是花了很多時間呢！所以請各位使用者，剛開始的學習曲線或許高了一點，但是只要耐心學習，會發現是很有趣的唷！那麼要如何解決問題呢？很多人直接就會上各大討論去問，當然問問題是很好的，這是可以學習和互相討論切磋的方式。但是很多常見的問題是大家已經討論很多遍了，所以上討論區發問以前，請先搜尋看看這個問題有沒有人討論過，有沒有解答。若真的找不到在去討論區詢問。而在詢問的方法上，請你把問題描述清楚，不要問那種籠統或者大家都搞不清楚你問題的詢問。

例如你問個怎麼我的網路不能上，這個問題原因很多，可能是你網路卡不支援 Linux，IP 設定錯誤，網路 ISP 連線有問題等。但是如果你很簡單的問題說怎麼不能上了，最好是大家有辦法用猜的猜出是哪裡的問題！如果你講說你用 WIN XP 可以正常上網，那麼討論區上的義工們就可以知道網路是好的。如果你也講了網路卡型號，那大家也可以幫你查那塊網路卡有沒有被支援。如果有支援的話，就很明顯是設定的問題。記得，網友們幫助你解決問題是一分不拿的，所以你也不能奢望大家免費幫你什麼都弄到好吧！所以問問題前，一定要做一些功課，至少要可以清楚的描述問題，如果這些都做不到而想使用 GNU/Linux 的話，我認為是有九成不可能的！那剩下來的一成呢？就是花錢買服務了，現在很多公司都有提供到府解決 GNU/Linux 服務，簡單來說就是花錢消災啦！不過這就一點都不好玩了，不是嗎？以下提供一個解決問題的步驟流程。

- **提問以前的準備功課：**首先先向 Google 大神請教，通常我遇到問題有百分之九十都可以用 Google 找到答案。而向 Google 請教的方式是，如果有遇到錯誤訊息，把它複製貼上到瀏覽器，再用 Google 去查，通常都可以查到。如果是不知道怎樣使用這類的，比較困難一點，但是可以試著先搜尋相關軟體的名稱，看看有沒有人有完整教學，或者配合條件搜索，來找到你要的解答。
- **去各大討論區爬爬文：**如果向 Google 還查不到，那麼可以去 1.3.2 節中介紹的網站，在那爬爬文章，看有沒有類似的問題。如果以前已經有人討論過且已經有解答了，直接再問一次是很沒有禮貌的，也是很浪費大家的時間。

- **提出問題來吧！**如果前面的方式還是找不到解決方法的時候，就可以提出問題來啦！先把你系統的狀態描述清楚，例如用什麼樣的硬體，你使用的 GNU/Linux 版本，和一些系統紀錄檔和錯誤訊息。然後就等看看有沒有人給你相關的解答。當然這不一定會有，或者一定有正確的解答，因為別人可能沒有遇到這個問題。

所以用 GNU/Linux 難不難？不難，就看你的心態了！所以筆者希望各位看官可以多多實際操作，並且在網路上廣泛閱讀相關文件，這樣可以培養對於問題的看法和直覺。並且可以多上上討論區和人討論，如果遇到自己會的東西，也不需要藏私，多多幫人解答。這樣也可以教學相長，也會在回答的過程中，讓你的所學更加精煉！如此筆者期許各位讀者都可以成為自我學習和樂為助人的 Ubuntu 專家。

1.3.2 網站

網路上有許多不錯的 GNU/Linux 資料和討論區，為了方便讀者索引，我將各種網站資源分門別類的整理了一下。而讀者遇到了問題，也可以視問題的種類來選擇下面網站來搜尋答案。

中文相關網站

- Ubuntu 正體中文站 - <http://www.ubuntu.org.tw>
這裡是 Ubuntu 的社群中文討論區，裡面常常會有一些和 Ubuntu 的相關新聞，當然也有很多人在上面討論相關問題。而讀者可以透過 dbtsai 這個代號找到我。
- Ubuntu 正體中文 wiki - <http://wiki.ubuntu.org.tw>
維基百科這個架構在最近越來越紅了，它允許任意使用者都可以修改上面的文章，結合社群的力量，來整理出完整的文件。而這個站就是使用同一個架構。雖然這裡才剛開始耕耘，但是這裡是一個很自由的世界，期許讀者可以在上面獲得有用的資料，並且把自己的成果在上面分享開來！
- 非官方 Ubuntu 新手指南 - <http://www.ubuntu.org.tw/ubguide/>
裡面的表達方式有點像 FAQ，就是它會在前面列出一大堆問題，然後你按進去就會有解答。裡面的問題都是針對新手常常遇到的情況來回答。例如

要怎樣安裝影片播放程式，或者如何安裝 Skype 等等問題，也有教讀者怎樣安裝顯示卡驅動程式等。

- 摩托學園討論區 - <http://moto.debian.org.tw>
這裡是 Debian 社群的討論區，因為 Ubuntu 和 Debian 本是親戚，所以這裡也很值得一看唷！而且這裡面高手雲集，絕對可以讓你學到東西的！
- DebianWiki 中文站 - <http://wiki.debian.org.tw>
也是 wiki 架構，進去以後，點「社群共筆索引」，裡面有很多有用的資訊。尤其是伺服器應用上的設定方式更是相關，所以想要把 Ubuntu 當成伺服器使用的讀者，可以參考這裡有關設定。
- Debian 參考手冊 - <http://qref.sourceforge.net/>
裡面提供了大量的 Debian shell-command 實際應用，並且也包含了系統管理的概念與實作。裡面涵蓋的主題有 Debian 的基本概念，如何包裝軟體包，如何在指令模式下安裝軟體，並教導你如何自行編譯符合 Debian deb 的 Linux Kernel 核心軟體包，也有系統效能微調，和網路及程式設計等相關的廣泛題材。因為裡面所談的都是系統架構和比較進階的部份，所以內容都和實際在 Ubuntu 上的情形大同小異。如果讀者想要進一步了解你的 Ubuntu，可以試著去讀看看。而裡面有分很多國家語言，當然讀者就選繁體中文就好了，並且也提供網頁版和 PDF 版，所以讀者可以視需求選擇。
- 鳥哥的 Linux 私房菜 - <http://www.vbird.org>
這是鳥哥為了要防止自己忘掉所學而所做的筆記。因為他的筆記很仔細，所以對於初學者幫助很大，所以現在紅透半天了。但是他慷慨的把所有內容讓在網路上給人家參閱。而裡面對於一些基本指令的用法，都有詳細的介紹。例如 tar 這個壓縮指令他就有詳細寫出各個參數的意義。所以對文字模式還沒有概念的讀者，可以來這邊看看。或者已經知道那在幹嘛的，但是常常忘了小細節的人，也可以在這裡查到完整的用法。
- Study Area 酷！學園 - <http://www.study-area.org>
這裡成立的目的是幫一些新手從基礎概念學起，並且帶給讀者自己思考和學習的能力。而這裡的討論區 <http://phorum.study-area.org> 裡面有很多熱心助人的高手，所以有空可以來這裡多多和他們切磋啦！

英文相關網站

- Ubuntu 英文官方網站 - <http://www.ubuntu.com>
這不用多說吧！這是 Ubuntu 的官方網站，裡面會有相關於 Ubuntu 的最新消息和相關資源。
- Ubuntu 官方討論區 - <http://www.ubuntuforums.org>
如果不怕英文的話，可以到這裡參與討論。往往在中文討論區找不到的答案，有時候在這裡可以發現。而且這裡所討論的資訊，也是比中文的資訊還要快一些。
- Ubuntu 官方 wiki
著 Ubuntu 在國外的流行，社群發展也日益茁壯，因此這種以社群來建構文件的方式，在英文 Ubuntu wiki 文件的深度與廣度日益完整。讀者可以在這裡取得人家整理好最新的文件，尤其是當你不想用討論區上的數種方法，進而慢慢試出哪種是最佳方案，那麼可以來到這裡，取得完整的 HOWTO。而如果我要找 XGL 的安裝和使用方法，就搜尋 XGL，然後就可以找到相關文件，裡面會有詳細的步驟說明。

1.3.3 IRC 即時聊天室

IRC 的全名是 Internet Relay Chat，也就是網際網路中繼聊天室。它主要的功能是用在群體聊天，但它也支援一對一聊天的模式。它的運作方式是透過網路連線到一個 IRC 伺服器，通常是不需要註冊，但是你必須要有自己設定的暱稱。連線到伺服器以後，我們就可以加入自己喜歡的「頻道」(channels)來參與討論。通常大部分的自由軟體計劃的程式設計師都喜歡在 IRC 上聊天討論，所以在這裡可以直接和開發者有對話的管道。它比論壇還方便的地方是 IRC 是即時的，也就是不像論壇需要等人家回覆，所以討論起來效率是比較好的！

在 Linux 常用的客戶端程式有 Gaim 和 XChat 等，而 MSWindows 裡面大家常用的是 mIRC，在 GNU/Linux 裡面連 IRC 的方式將會在後面提及！

中文相關 IRC

- Ubuntu 台灣 IRC - 伺服器：irc.freenode.net 頻道：#ubuntu-tw
這裡是台灣 Ubuntu 社群所建立的 IRC，裡面討論主要是使用 UTF-8 編碼，所以別設錯編碼了，不然大家會看不懂你在聊什麼唷！
- Debian 台灣 IRC - 伺服器：irc.debian.org 頻道：#dot
是台灣 Debian 社群所建立的 IRC，也是採用 UTF-8 編碼，這裡高手雲集，看看別人在聊什麼都很值得。

英文相關 IRC

- Ubuntu 官方 IRC - 伺服器：irc.freenode.net 頻道：#ubuntu
這是 Ubuntu 官方的 IRC 聊天室，當然一些 Ubuntu 的開發者也會在這裡流連了。
- Debian 官方 IRC - 伺服器：irc.debian.org 頻道：#debian
這是 Debian 官方的 IRC 聊天室，上面人氣很盛，一些開發者都會掛在上面。

1.3.4 如何提報錯誤

如果真的遇到是軟體錯誤，當然要提報給開發者修正啦！這有幾個管道，你可以提報給最原始的程式設計師，或者是提報給軟體的包裝者。有時候程式碼是正確的，但是問題是因為軟體包裝者包裝錯誤，所以這時候就算回報給程式設計師也沒用吧！所以遇到這方面的 bug，可以先上 IRC 或討論區和別人討論一下，確定並且釐清問題的原因後，你就可以透過各種管道來反應錯誤！

向套件的包裝者提報錯誤，如果是因為原始程式碼有問題，通常套件包裝者會幫忙代為通知或給與修正。而如果是因為包裝時有錯誤，他也可以給與更正。而 Ubuntu 為了簡化這些提報流程，特別成立了 <https://launchpad.net> 這個網站。他整合了提報錯誤，翻譯和程式碼開發等單一整合窗口，也就能簡化研發流程。而如果你確定是有 bug 的話，你必須先註冊一個帳號，然後登入後，選擇你遇到有錯誤的套件，並且把詳細的錯誤描述出來。你可以多找幾個人去複證錯誤，那樣開發者會把它移到比較優先處理的錯誤。

還有一種方式是可以寫信給 package 維護者，你只要下以下指令

```
$ apt-cache show <${package}>$
```

那就可以查詢到 maintainer 的 E-mail，所以你也可以寫信給他。這個方式提報的對象和上面相同，都是軟體包的包裝者，也就是 Ubuntu 或者是 Debian 的開發者。

當然如果你確定是原始程式碼就有 bug，你也可以直接寫信給原始程式設計師，通常他的 E-mail 可以在網路或軟體手冊上很容易找到。

1.3.5 聯絡筆者

這本書是定位給初學者或者中階讀者作為 Ubuntu 入門使用，所以筆者也囉唆了一點，講很多故事，希望讀者能從根本瞭解，而不是只有硬生生的指令，照著打而導致不知其所以然。希望筆者把背後的人文背景介紹給你們，可以讓讀者體會自由軟體的精髓，進而有能力自我精進。但是東西一多，難免會有錯誤或者辭不達意，因此讀者如果對書裡面的內容有問題的話，歡迎寫信給筆者討論，並且也歡迎讀者的批評和指教。

- 蔡東邦 - dbtsai@gmail.com
- 李聖璋 - ducati@ubuntu.org.tw

除此之外，本書的勘誤表以及最新資訊可以在下面所列之網址找到：

- <http://www.ubuntu.org.tw/modules/newbb/viewforum.php?forum=25>

CHAPTER 2

安裝前準備

相信各位讀者看到這裡，一定迫不及待想嘗試看看了吧！但是筆者還是得要稍微介紹一下硬體需求、基本的 GNU/Linux 知識和如何取得光碟等，這些都會在安裝的過程中派上用場喔！所以這一章可以看成是預備知識，要教大家怎麼樣辨別硬體是否支援、該如何去挑選一個有支援的 3D 加速卡，或者選一個在 GNU/Linux 底下可以使用的無線網卡。其中也會談到一些基礎觀念，例如什麼是分割區，或者分割區所使用的檔案系統格式其效能好壞等觀念。筆者建議若讀者已經具備 Linux 安裝的基礎知識則可以略過。

2.1 硬體需求

玩 GNU/Linux 還要挑硬體喔？！其實大部分的問題都出在顯示卡和無線網路卡上面，其他大部分的設備 GNU/Linux 幾乎是都有支援的。所以我把顯示卡和無線網路卡的部份在此獨立出來，希望能提供讀者容易比較、選購。其實基本上相容性的問題與品牌的關係不大，而是在硬體所使用的晶片組。例如在不同品牌的顯示卡上，只要都是用 Nvidia 的晶片，則驅動程式都可以共用。相同的概念可以用在其他的硬體上，只要是相同晶片的無線網路卡，都可以用一樣的驅動程式驅動。所以選購相容於 GNU/Linux 的硬體，應該要選擇與確認的是設備所使用的晶片有支援，而不是在於某個品牌一定會和 GNU/Linux 相容唷！

2.1.1 主機硬體

首先，為了確定 Ubuntu 是否可以在你的機器上順利工作，在安裝前先瞭解你的主機硬體是否支援是很重要的。主機硬體不外乎是以下架構，我們將會一一個別討論和建議使用者最低的安裝需求。

- **CPU**：使用建議 Intel Pentium 4 1.6GHz 或 AMD K7 1.4GHz 以上：目前還沒聽說過有 GNU/Linux 不支援的 x86 CPU，不管是雙核或是多執行緒（HT）的 CPU，只要是市面上買的到的 586 以上相容的 CPU，應該都可以安裝 GNU/Linux！其中差別只在快慢而已。而事實上如果是在伺服器任務上使用的话，Pentium 3 500MHz 其實就很夠用了，所以 CPU 選擇主要是依你要在該系統平台上做些什麼任務而定！筆者曾經看過賽陽 400 的 Notebook 也能跑得很開心，所以不是沒達到建議速度的 CPU 就一定不能跑，只是比較慢一點而已啦！
- **RAM**：使用建議有 512 MB 以上：記憶體當然越大越好啦！因為當記憶體不足的時候，系統將會使用虛擬記憶體，也就是會用硬碟來模擬記憶體。這個時候你會發現電腦開始像烏龜一樣慢，硬碟會一直嘎嘎地叫呢！
- **主機板**：只要配合你的 CPU 選購即可：通常主機板的晶片組相容性都很好，所以挑選時只要選擇穩定性佳和維修方便的品牌與型號。有時候主機板在 GNU/Linux 上驅動會比在 MicroSoft Windows 方便，因為在 MicroSoft Windows 中，使用者甚至需要自行安裝 USB、內建網卡與 SATA 硬碟控制卡等驅動程式才能順利使用，而通常這些驅動程式在 GNU/Linux 的核心上都已經內建了。至於在主機板上會遇到的相容性問題通常都會出在內建磁碟陣列（RAID）卡上，這將會在磁碟控制卡那個章節一起討論。
- **顯示卡**：我們將會在下一節討論。
- **光碟機**：隨你高興啦！只要能正常讀寫光碟片的就可以安裝了：如果有 DVD 就更方便了，因為 Ubuntu 也有 DVD 版本，此時安裝的時候就不用再上網抓取額外的安裝套件。不過如果你的網路環境的條件不錯，那可以用一般 CD 版本的安裝開機片安裝，之後再上網去抓取一些中文使用的套件。
- **硬碟**：約 40 GB 左右的硬碟就很夠用了：目前在 Ubuntu 上可以玩的套件愈來愈多，所以硬碟大一點當然可以玩比較多東西。但不一定得要 40 GB

才夠，像筆者的筆記型電腦在系統部份也只用掉了大約 10GB 的硬碟空間而已，所以請視你本身的需要來決定。

- **網路卡**：目前主機板上內建的都很好用，不論是 100Mbps 或 1Gbps，幾乎所有晶片組或廠牌的網路卡都可以在 GNU/Linux 上使用，並且不需要安裝驅動程式。貴的可到上萬元一片的 server 專用光纖網卡，或便宜到一片兩三百元的螃蟹晶片網卡，筆者都有使用過的經驗，基本上這項設備很少有相容性問題！假如你是當成桌面使用，內建網卡就夠用了。如果你的機器比較舊，沒有內建網卡，那麼筆者推薦你加裝螃蟹晶片網卡，這張卡到各大電腦公司買的話，幾百元就有了，俗擱大碗。但如果你是要在 server 上使用，建議是採用比較穩定的 Intel 或 3Com 出品的網卡。
- **無線網路卡**：我們將會在後面專節討論。
- **磁碟控制卡（RAID、SCSI、SATA 等）**：這項設備最大的問題是廠商往往為了節省成本，主機板上內建的 SATA 或 IDE 控制模組所附的 RAID（磁碟陣列）系統其實是「假的」，這些卡是透過驅動程式去模擬磁碟陣列的動作，使用主機板上的 CPU 來計算 RAID 陣列資料，而不是真的在硬體上就有處理器去做 RAID 運算。所以很多 GNU/Linux 的使用者也許會疑惑，為什麼在 BIOS 設定好的 RAID 在 GNU/Linux 上辨認不出來呢？原因就是這些內建 RAID 陣列都是模擬的，因此要透過特殊的驅動程式去模擬使用，因此如果沒有經過特殊設定，當然在 GNU/Linux 上就不能正常使用啦！所以如果打算在這些「假的」RAID 設備中使用 RAID 功能，建議直接使用 GNU/Linux 核心（Kernel）的軟體 RAID 功能，把這些設備當成一般控制卡使用就好了。如果要在 GNU/Linux 上玩硬體陣列，筆者建議買那些高檔的 SATA 或 SCSI RAID 控制卡，這些通常都有提供 RAID 0、1、0+1、5 等等的陣列功能，並且上面都有專屬的 CPU 計算陣列資料。當然這些卡都不便宜，可能一片都要上萬元，不過在整體效能上一定比軟體陣列好很多，因此推薦伺服器使用者使用。
- **音效卡**：主機板上內建的 AC97 就很不錯用囉！一般主機板上內建的 AC97 音效卡都可以「無痛」在 Ubuntu 上使用。但如果你對音質很著重的話，則可以自行安裝好一點的音效卡一般外接音效卡都可以順利的被 GNU/Linux 偵測到。若你真的想要選購的話，可以到 ALSA（<http://www.alsa-project.org>）網站去查詢有支援的音效卡。
- **鍵盤和滑鼠**：就找一個你最順手的吧！

在此做一個小總結，就是 GNU/Linux 是不大挑硬體的，不論是自己組裝的電腦或外面買的現成套裝電腦，基本上都應該沒有問題。所以讀者你的桌上型電腦，應該是可以不需經過任何硬體上的調整，就可以順利安裝 Ubuntu。

至於在筆記型電腦上，大部分的數據機（Modem）是不能使用的，因為在筆記型電腦上的數據機大部份是軟體模擬的。而像電源節省模式或是電池狀態查詢等，則都應該是可以正常使用的。如果讀者想要查詢你筆記型電腦的 Linux 支援程度，可以到 Linux on Laptops（<http://www.linux-laptop.net/>）網站查看別人安裝在與你同款筆記型電腦的心得和所遇到的問題。筆記型電腦的無線網路問題，筆者會在後面無線網路卡的部份一起討論。

2.1.2 支援 OpenGL 加速的顯示卡

隨著 GNU/Linux 的 3D 遊戲盛行和 AIGLX 的到來，以及未來 cairo 將整合向量繪圖與 OpenGL 加速，GNU/Linux 上也越來越需要 3D 和 2D 的硬體加速。在 GNU/Linux 中最常用的繪圖 Application Programming Interface（API）是 OpenGL，所以我們當然希望挑選到一塊對 OpenGL 加速有良好支援的顯示卡啦！目前支援 GNU/Linux 比較好的顯示卡都是採用 Nvidia 和 AMD（ATI）所生產的晶片。並非其他廠商的卡不能在 GNU/Linux 上使用，而是那些卡通常都沒有提供 GNU/Linux 的 3D 加速驅動程式，所以造成它只能當成一般的顯示卡使用。



圖 2.1 Nvidia Logo

目前 Nvidia 有支援 GNU/Linux 的晶片型號是 GeForce 全系列和 Quadro 全系列。因此在購買的時候，只要認上圖 2.1 的 logo 即可，至於製造的廠商就依你的喜好選擇了！

備註 你可以參考 http://www.nvidia.com/object/IO_18897.html 取得更詳細的支援列表。)



圖 2.2 AMD (ATI) Logo

在 AMD (ATI) 這家公司對於 GNU/Linux 的支援方面，目前提供 Radeon 8500 和 FireGL 8700 以後的所有顯示卡的驅動支援，也就是早於 Radeon 8500 或 FireGL 8700 所生產的晶片將不會支援。而選購方式和 Nvidia 相同，只要認上圖 2.2 AMD (ATI) 的 logo 就可以了，因為在 GNU/Linux 中相容性的關鍵是所使用的晶片，而不是生產的廠商。

2.1.3 目前原生支援的無線網路卡

目前無線網路卡在市面上有三種規格，分別是 802.11a，802.11b 和 802.11g。802.11b 的傳輸速度是 11Mbps，而 802.11g 和 802.11a 都是 54Mbps 傳輸速度的規格。由於 802.11b 和 802.11g 都是採用 2.4GHz 的無線頻段，所以互相可以相容。而 802.11a 則採用 5GHz，所以不能和其他的規格相容，因此在台灣使用率不高，建議使用者不用考慮 802.11a 了！

在目前筆記型電腦內建的無線網卡方面，若是 Intel Centrino 晶片組系列的筆記型電腦，通常內建的無線網路卡在驅動上都沒有問題，但少數 Centrino 的筆記型電腦沒有採用 Intel 的配套 Intel PRO/Wireless 無線網卡，這個時候就要查詢一下是用什麼晶片。若你的筆記型電腦確定是採用 Intel PRO/Wireless 無線網卡，則無線網路對你來說已經不構成問題了，可以跳過這一小節囉！

反之，若非使用此系列晶片組的筆記型電腦，還是有可能被 Ubuntu 支援，讀者可以透過指令 `lspci -v` 去察看有 wireless 關鍵字的那幾行，會有所使用晶片的詳細資訊。然後讀者可以在接下來的部份讀到各種晶片支援的訊息。若是舊款的筆記型電腦沒有內建無線網卡，而讀者想要外加無線網卡，那同樣你也可以依照下面的資訊去選擇購買。

目前在 GNU/Linux 上驅動無線網路卡的方式有兩種：一種是透過 `ndiswrapper` (<http://ndiswrapper.sourceforge.net>) 去載入在 Microsoft Windows 上的驅動

程式，這樣幾乎所有的無線網路卡都可以驅動，因為無線網卡總不能連 MS 的驅動程式都不提供吧，那可就不用賣了！但是這樣會有幾個缺點，例如無法在 64 位元環境或非 x86 環境下使用，並且因為這不是 GNU/Linux 原生的驅動方式，所以效能和功能會有所缺憾。當然，若你的無線網卡真的沒有被支援的話，這就是不得已的方法了！

第二種方式則是透過 GNU/Linux 核心 (Kernel) 內建的驅動程式原生的支援，這是最理想的方式，也是能跨越不同架構來驅動無線網卡的唯一方式。然而，無線網卡的晶片廠商為了競爭而留一手或者因為暗藏一些用軟體模擬的部份若公開的話會破壞電信法規，因而導致廠商不願意釋出規格或原始碼來提供 Linux hacker 撰寫驅動程式，這迫使在 GNU/Linux 上無線網路原生支援成為一個很大的問題。所以目前在 GNU/Linux 上很多無線網路驅動程式是用反組譯的逆向工程弄出來的。當然，也是有一些對於 GNU/Linux 友善的晶片廠商會釋出規格讓 hacker 們撰寫驅動程式。以下筆者將談及目前可以在 GNU/Linux 下原生驅動的無線網路晶片。

無線網卡的廠商一般而言並不會像顯示卡廠商一樣地標出所使用的晶片，所以我們必須要自行測試收集各個無線網卡所使用的晶片，而筆者會提供一些網站，供給讀者選購的時候當成參考。當然最簡單的方式是讀者帶著你的電腦直接去電腦公司，然後一片一片的把卡插上去試試看，這樣就能百分之百確認哪一片卡能用了！

- **Atheros**: <http://madwifi.org>: 這款晶片的驅動程式已經內建在 Linux kernel 裡面了，這款晶片在 Ubuntu 的支援性不錯。而它是由 Atheros 公司所生產。這個系列的晶片分別支援了 802.11a/b/g 三種規格。
- **Atmel AT76C50X**: <http://atmelwlandriver.sourceforge.net>: 這款晶片在 Linux kernel 內也已經支援了，不過這一系列的卡只有 802.11b 的規格。如果不嫌它速度慢和新品難買的話，它還是屬於很好用的無線網卡。
- **Broadcom BCM43xx**: <http://bcm43xx.berlios.de>: 這個驅動程式最有名的地方是在這顆晶片被廣泛地用在蘋果電腦 (Apple) 上。但是這些機器並不是 x86 的平台，所以沒辦法使用 ndiswrapper 去載入 MS 的驅動程式。這激怒了一堆駭客，終於用逆向工程破解了內部結構。kernel 2.6.17 以上版本包含了這支驅動程式，因此在 Ubuntu 上也可以順利的使用了。這系列晶片目前也有 802.11a/b/g 三種支援。

- **Intel PRO/Wireless** : <http://ipw2200.sourceforge.net> : 這系列無線網卡有 Intel PRO/Wireless 2100, 2200 和 2915ABG。其中 2100 是支援 802.11b 的無線網卡, 2200 支援 802.11b/g 兩種, 2915ABG 支援三種規格 802.11a/b/g。通常是搭配 Intel Centrino 晶片組的筆記型電腦出貨, 但並不一定 Centrino 晶片組的筆記型電腦就是搭配這幾種無線網卡。而這幾種無線網卡目前在 GNU/Linux kernel 上都有很好的支援, 因此使用這些晶片的筆記型電腦都可以很容易的在 Ubuntu 的環境下無線上網。
- **Orinoco** : http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Orinoco.html : 這款晶片的驅動程式在 GNU/Linux kernel 內也是已經支援了, 不過這一系列的卡只有 802.11b 的規格。因此如果不嫌它速度慢和幾乎買不到新品了, 它還是很好用的無線網卡。
- **Prism 2, 2.5, 3, GT** : <http://www.linux-wlan.org> : 這一系列晶片所支援的規格為 802.11b, 而這款的驅動程式已經在 GNU/Linux kernel 裡面內建, 所以不需要安裝, 只要設備插上去就可以運作了。
- **Prism 54** : <http://prism54.org> : 如同其名, 這一系列都是支援 802.11g 54Mbps 的無線網卡晶片。而這款驅動程式也是已經在 GNU/Linux kernel 裡面內建了。
- **Ralink RT2x00** : <http://rt2x00.serialmonkey.com> : 這是一間台灣的無線網路晶片公司。他們公司對於 GNU/Linux 有親切的態度, 但是公司官方的驅動程式臭蟲很多, 沒有達到實用的地步。因為官方驅動程式以 GPL 授權, 現在社群正在全力的幫忙改善, 目前社群版的驅動程式已達到堪用的地步, 並且準備未來放入 kernel 內, 那麼使用者就不需要再自己編譯驅動程式。這系列晶片目前有 802.11b/g 兩種支援。

備註 目前一般使用要成功驅動還有點困難, 需要自己嘗試編譯一下才能成功安裝。

- **TI ACX100/111** : <http://acx100.sourceforge.net> : 這是德州儀器的晶片, 有 802.11g 和 802.11b 兩種不同規格。目前 Ubuntu 已經支援它了。
- **Zydas** : <http://zd1211.ath.cx> : 這是台灣的公司, 益勤科技。它也有 Open Source 的驅動程式。這系列晶片目前有 802.11a/b/g 三種支援, 在 Ubuntu 裡面可以直接使用了唷!

還有一個麻煩的問題，就是出在無線網卡的韌體（firmware）。一般而言，韌體裡面存放一些控制該外部裝置的程式，也就是一般的外部硬體會有自己的處理器來作一些硬體上操作與運算，而韌體就是負責這件事的程式。因此大部份裝置的韌體會儲存在該硬體的快閃記憶體（flash memory）中，所以如果我們要更新的話，就需要透過特殊的軟體去更改那個嵌入在硬體上的軟體。

但是目前很多無線網路卡為了降低成本，都會把快閃記憶體省略掉，所以我們需要在驅動裝置以前，把韌體從檔案載入到系統給該裝置使用。故我們在使用無線網卡時，就得要自己另外提供韌體才能使用。當然有一些是直接把韌體燒在硬體上的無線網卡，那就不會有這種煩惱啦！

有些公司的韌體授權是可以讓 Ubuntu 包含該韌體在系統裡面，例如 Intel PRO/Wireless 系列、Zydas 系列、Atmel 系列、prism 54 和 TI 系列，Ubuntu 都有提供韌體放在 /lib/firmware 裡面。所以使用這幾種晶片的無線網卡就不需要自己去下載韌體，會比較方便。例如像 bcm43xx 系列就要自己另外上網去下載。

上面講了林林總總的晶片組，但如果我們在賣場買無線網卡時卻無法得知那是用什麼樣的晶片，那不就是白說了嘛？其實這些是為了下面的討論作準備的，下面有幾個參考網站，可以用晶片查廠商型號或用廠商型號查所使用的晶片。所以建議大家先到購物網站上鎖定幾款型號，再進一步查看所使用的晶片是否相容。

- **Ubuntu 中的無線網卡實測**：<https://help.ubuntu.com/community/WifiDocs/WirelessCardsSupported>：列出了目前在 Ubuntu 上經過實測的無線網卡廠牌和型號。注意「Supports network install?」那一欄，如果是 Y 的話，代表可以在光碟開機後自動載入韌體並驅動，也代表了是系統內建韌體的款式，所以在相容性上比較沒有問題。這可以當成一個很好的指標。
- **Linux wireless LAN support**：<http://linux-wless.passys.nl>：無線網卡型號或晶片是否支援 GNU/Linux 可以在這個網站上查詢到，而且在這裡可以對個別晶片或廠牌作交叉查詢，其中標記為綠色的就是經過測試、確定可以支援原生驅動的無線網卡，讀者可以在購買前依據這些資訊挑選。而筆者的建議是挑選那些晶片的韌體已經進入 Ubuntu 官方的，那麼你遇到的問題會比較少。

- **韌體資訊**：<https://help.ubuntu.com/community/WifiDocs/WirelessFirmware>：列出目前在 Ubuntu 裡面有內建韌體的晶片款式，可以配合上面的網址用晶片選擇要購買的網卡廠牌和型號。
- **各種晶片在 Ubuntu 上的資訊**：<https://wiki.ubuntu.com/WirelessChipsets>：這裡包含了一些晶片的支援資訊，甚至提供非原生驅動（透過 ndiswrapper）是否能成功的資訊。

2.2 取得光碟

在這裡筆者要教大家釣魚的方法，可以讓讀者隨著版本更新，不需要再購買書籍就可以取得最新版的安裝光碟。



圖 2.3 Ubuntu 原廠官方光碟

說明	網址
官方 mirror 下載點一覽表	http://www.ubuntu.com/download
官方總站下載點 (包含 DVD 和 CD)	http://cdimage.ubuntulinux.org
新竹縣網中心網頁 CD 下載點	http://apt.nc.hcc.edu.tw/pub/ubuntu-cd/
國立暨南大學網頁 CD 下載點	http://ftp.ncnu.edu.tw/Linux/ubuntu/ubuntu-cd/
元智大學網頁 CD 下載點	http://ftp.cse.yzu.edu.tw/ftp/Linux/Ubuntu/ubuntu-cd/
美國 DVD 下載點	http://mirror.mcs.anl.gov/pub/ubuntu-iso/DVDs/ubuntu/

在 Windows 平台下，直接用瀏覽器從網頁下載光碟映像檔，容易檔案不完全，或者斷線而需要重新下載。因此如果要直接從網頁下載，最好準備續傳軟體。若沒有續傳軟體的話，我們可以透過 FTP 這種專門抓檔案的方式來取得映像檔哩！在 Windows 平台的話，建議你用 FileZilla (<http://filezilla.sourceforge.net>) 來下載映像檔，這套下載點可以在官方網頁找到，或者國內很多網站也可以取得。

台灣有幾個官方承認的 mirror 站可以下載到光碟映像檔，可惜這些站並不提供 DVD 版下載，若要下載 DVD 版的，需要到國外網站下載。這幾個 mirror 站除了提供網頁模式下載以外，也提供 ftp 模式下載，若要用 ftp 模式的話，把網頁下載點網址的 http 和最後面的部份去掉，就是 ftp 位址了。例如要用 ftp 下載 iso 的方式是在 FileZilla 裡面位址的地方輸入 apt.nc.hcc.edu.tw，然後按連線，右邊遠端目錄進去 ubuntu-cd 裡面，再進入最新版本 6.10 的目錄。裡面有很多的 iso 檔，記不記得我們提過的，安裝光碟有分 Desktop 和 alternAMD (ATI) ve 版？前者是 LiveCD 加安裝光碟，後者是文字介面的安裝光碟。記得要選擇 368 版本的，然後把檔案從右邊拉到左邊本地目錄，過一陣子後，就下載完啦！

讀者也可以用續傳軟體 FlashGet 下載，這是一個可以免費試用的商業軟體，讀者可以到 <http://www.flashget.com> 來下載安裝！首先先用瀏覽器開下載網頁，讀者視需要哪一個 iso 映像檔，記得先不要用瀏覽器抓，而是把它拖到 FlashGet 的下載區，這樣就開始抓了唷，它預設放在 c:\downloads 裡面！

若是在 Linux 上就更方便了，你就到網頁上，在要下載的鏈結處，按右鍵選擇複製鏈結網址。然後在 Linux 的終端機下，用 wget 貼上網址，就可以下載啦！

Linux 下載範例：

```
dbtsai@ubuntu:~$ wget
http://apt.nc.hcc.edu.tw/pub/ubuntu-cd/6.06/ubuntu-6.06-desktop-i386.iso
--04:19:11--
http://apt.nc.hcc.edu.tw/pub/ubuntu-cd/6.06/ubuntu-6.06-desktop-i386.iso
=> `ubuntu-6.06-desktop-i386.iso'
正在查找主機 apt.nc.hcc.edu.tw... 163.19.1.81
正在連接 apt.nc.hcc.edu.tw|163.19.1.81|:80... 連上了。
已送出 HTTP 要求，正在等候回應... 200 OK
長度: 731,744,256 (698M) [text/plain]

0% [          ] 1,100,684  126.18K/s  ETA 1:43:32
```


取得 iso 映像檔後，讀者可以用 md5sum 去驗證抓取的 iso 檔是否有問題。在 Windows 平台上可以用 WinMD5 來計算 iso 的 MD5 hash，這支程式可以在 <http://ftp.isu.edu.tw/pub/Hardware/tools/Utility/WinMD5> 下載。用這個程式計算你下載的 iso MD5 hash 以後，可以再和網頁上公佈的 MD5 值比較，若相同的話就是下載沒問題了。在 Linux 底下就直接打指令 md5sum 加上檔名，就會顯示出 md5 hash 的值。當然這個步驟略過也可以，不過為了保險起見，還是驗證一下吧！

最後在 Windows 上可以使用 Nero 來把 iso 檔燒成光碟，在 Linux 上可以用 K3B 來達成。這對要玩 Ubuntu 的你，下載和燒錄光碟應該都不成問題吧！不管怎樣，最新版的 6.10 (Edgy) 安裝光碟是隨書附送，你暫時還不用煩惱這個問題。

2.3 安裝前規劃系統的預備知識

由於我們整個作業系統都將安裝在硬碟上，包含所有資料、文件等也都是存放在硬碟上，所以一個正確的硬碟分割區規劃可以幫助救回損毀的分割區。而正確的選擇格式可以使檔案讀取效能增加，同時檔案系統浪費的空間也會減少。最重要的是，可以確保檔案系統的穩定。

在介紹分割區之前，讀者還需要稍微瞭解一下 GNU/Linux 的目錄架構，因為 GNU/Linux 不像 Microsoft Windows 一樣是分割成 C、D、E 等磁碟槽，而是將所有的分割區都掛載在同一個 / (root) 目錄底下的某個子目錄，所以要先稍微介紹一下它的目錄架構。但這樣還不夠，由於每個分割區都是由硬碟所切割而成，所以當然也要先介紹硬體設備在 GNU/Linux 的命名與概念。

所以我們接下來將要介紹硬體設備的代號、目錄架構、和什麼是分割區以及各種檔案系統的優缺點，最後再帶大家進入實務上如何處理與規劃。

當然，GNU/Linux 也可以讓你完全不需要懂這些就可以安裝來玩玩看，如同 Microsoft Windows，你也可以只分割出一個分割區，然後全部設定都給系統核心自動設定，不過對於進階讀者來說，應該還是會想要由自己掌控吧！這樣才有成就感呀！

雖然下面的知識有點難、複雜，但這是學好 GNU/Linux 最基本與必經的過程。所以請大家慢慢地讀懂、讀通，若第一次讀不懂也沒關係，可以先略過，不用一直卡在這裡。等你安裝好系統後，當哪天有使用到相關的知識時，再回來閱讀。或者無聊的時候，也可以拿出來翻一翻哩！

2.3.1 GNU/Linux 目錄的架構

在 GNU/Linux 裡面，所有的目錄都是從 / (root) 根目錄所延伸的樹狀架構。也許有讀者會問，如果有很多顆硬碟，那不就有很多根目錄嗎？非也非也！其實那些硬碟的某個分割區不會像 MS Windows 一樣，分割成 C 槽、D 槽和 E 槽等等，而都是在那個根目錄或底下的某個子目錄，只是把某個分割區掛載 (mount) 到那個目錄罷了，也就是說在 Microsoft Windows 的 X 槽對應到 GNU/Linux 的話，就是掛載在某一個目錄底下。

那麼這樣子掛來掛去，總要有個起始源頭讓我們掛吧？所以至少要有一塊分割區給根目錄 / (root)，其他的目錄就會以此為基礎陸續掛在根目錄的某個子目錄下唷！如此就會形成一個如圖 2.4 的樹狀結構。

由於 GNU/Linux 有太多發行版了，但在早期每個發行版都有自己偏好的目錄配置方式，因此早期造成很多使用者的困擾。例如某個使用者在 A 發行版上跑一些程式，結果哪天他換 B 發行版了，但是他可能在 B 發行版一樣的位置找不到該程式所使用的函式庫。為了平息這樣的紛爭，Filesystem Hierarchy Standard (FHS ; <http://www.pathname.com/fhs/>) 就出面制定了一些標準，讓各個發行版有個綱要可以知道哪個目錄要放什麼樣的東西。目前 FHS 定義的是每個目錄的大綱，所以很多細部設定檔的檔名或許還是有所不同，但是幾乎都放在同一個目錄底下了。這樣使用者在尋找的時候，應該是容易許多了。所以 Ubuntu 也遵循這樣的命名結構，圖 2.4 在別的發行版也適用哦！

在瞭解了上面的內容之後，我們知道安裝時一定要有一個 / (root) 根目錄，其他目錄是依附在它下面。這些其他的目錄所使用的實體硬碟空間可以是獨立出來在另外一顆硬碟或同一顆的其他分割區或者甚至不用獨立出來。如果沒有獨立出來的話，就是使用 / (root) 根目錄分割區的空间或者在上一層的實體分割區空間。

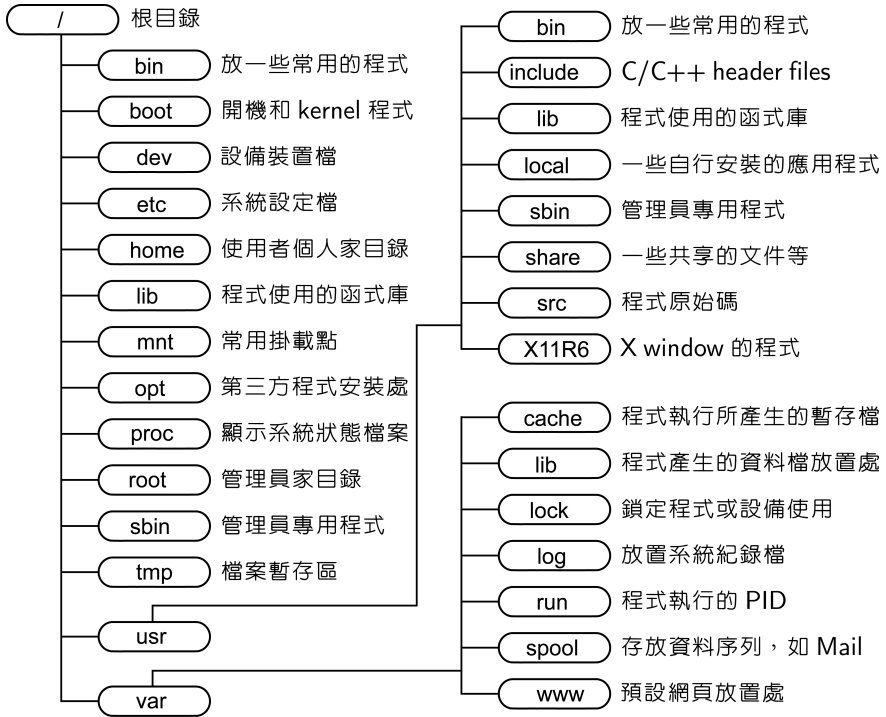


圖 2.4 Ubuntu 目錄架構示意圖

根據表 2.1，讀者可以自行評估哪一個目錄比較重要，然後決定是否獨立出來，這樣子可以拿來當成損害控管之用。後面也會有一些比較容易看懂的實際例子。

表 2.1 GNU/Linux 特別需要注意的目錄

目錄	功用說明
/boot	這裡面放了開機所需要的檔案，例如 Linux kernel image 和一些設定檔等。開機時，載入開機管理程式 (bootloader) 之後，就會掛載這個目錄並把核心映像檔載入到記憶體。所以如果這個分割區的格式是 bootloader 認不得的話，就會發生很嚴重的問題。最保險的是這個目錄採用 Ext3 或 ReiserFS 的分割區格式。
/etc	整個系統的設定檔幾乎都放在這個目錄底下。所以要備份時，這個目錄千萬可不要忘了唷！

目錄	功用說明
/home	這裡放了使用者所有的個人檔案，例如你的照片啦，影片和文件等，也就是所謂的家目錄。所以這個目錄很重要唷！建議你把它分開到額外的分割區，萬一那天系統掛了還可以保留你的文件喔！
/var/log	所有系統登入或一些系統狀態的資訊都會紀錄在這裡。所以如果你是用在伺服器上的話，可以單獨分割成一個分割區給它，避免因為太多的服務紀錄檔而把硬碟的容量塞爆喔！一旦被塞爆，系統運作就會不正常了。若是在桌面系統的應用上的話，就沒有這個考量啦！
/var/spool	這裡常常會被 E-Mail 伺服器當成佇列資料存放的地方。所以用在 E-Mail 伺服器的情形下，建議還是獨立到額外的分割區比較好。

2.3.2 硬體設備的代號

在 GNU/Linux 裡面有個很好玩的特色，就是所有硬體設備都是可以看成是某個檔案來操作。這樣有個好處是，應用軟體只要管好對那個檔案要如何操作，而不必要管實際上硬體的運作。其中代表設備的檔案和實際運作的硬體中間的關連，就是 GNU/Linux kernel 所需要負責的。

對於 IDE 硬碟來說，常常組裝電腦的人應該都知道主機板上最多只有兩個 IDE 插槽，而每個插槽都可以透過排線接兩個硬碟吧！主機板上第一個插槽就是 IDE1 (Primary)，第二個插槽就是 IDE2 (Secondary)，每個插槽上可以接的兩顆硬碟其中一顆是 Master，另外一顆就是 Slave，至於怎樣判斷可以透過硬碟上的 jumper 來得知。若是硬碟上的 jumper 插在 cable selection 的話，那麼從排線前端數過來第一顆就是 Master 啦！如表 2.3，我們可以得知插在第一個插槽的 Master 硬碟，它的檔案代號位置就是 /dev/hda！若光碟機是裝在第二個插槽且 jumper 預設是 Slave 的話，這時候光碟機的檔案代號的位置就在 /dev/hdd 喔！

對於 SCSI、SATA 和隨身碟則都是 /dev/sd[a-z]，但是它們會依照順序排下去，理論上會先排 SCSI，然後是 SATA，最後才是隨身碟，不會有重疊的部份。因為現在很多新的電腦都採用 SATA 硬碟，所以請記得這些 SATA 硬碟的代號都是和 SCSI 用一樣的唷！舉例來說，若你有兩台 SATA 硬碟，但是沒有 SCSI 裝置，那麼第一顆 SATA 硬碟的檔案位置代號就是 /dev/sda，第二顆就是 /dev/sdb，很簡單吧！

表 2.2 GNU/Linux 設備檔案位置

設備	該設備檔案代號位置
IDE 硬碟	/dev/hd[a-z]
SCSI 硬碟機	/dev/sd[a-z]
SATA 硬碟機	/dev/sd[a-z]
隨身碟或讀卡機	/dev/sd[a-z]
光碟機	/dev/cdrom (通常是 soft link 到 /dev/hdd)
磁帶機	/dev/st0
軟碟機	/dev/fd[0-1]

表 2.3 GNU/Linux IDE 硬碟代號

IDE \ Jumper	Master	Slave
IDE1 (Primary)	/dev/hda	/dev/hdb
IDE2 (Secondary)	/dev/hdc	/dev/hdd

2.3.3 分割區簡介 (Partition)

前面已經提到了，在 MicroSoft Winsows 中同一顆硬碟的各個槽其實就代表了不同的分割區。而常常一般人喜歡把硬碟分割成 C 槽和 D 槽，此時若需要重灌時所有資料和文件都放在 D 槽，就可以只把系統碟 C 槽格式化刪除，再重灌系統，如此資料就不用備份出來，可以很容易完成重灌且保存資料！同樣地，在 GNU/Linux 也是一樣，我們喜歡把重要的目錄分別放在不同的分割區，這樣子系統出了問題也能很容易來救援。

或許上面的講法太過於抽象，我們可以想成一顆新硬碟就像是一間剛蓋好的大辦公室，裡面沒有隔間，所以整個空間很大，也許有 100 坪。但其實這樣的空間並不方便被使用，我們需要根據業務的不同把整個空間作隔間，可能隔成客服部門 30 坪，技術部門有 40 坪，管理階級辦公室 30 坪。彼此之間都有隔間，這樣就算某個部門發生火災，其他部門也不會受影響喔！想像一下，經過一陣子的辦公後，各部門的傢俱和裝潢都已經就定位了，如果這時候才要改變坪數規劃，是不是很麻煩呢？所以電腦硬碟的分割區也是相同的道理！

現在我們要開始講磁碟分割表（partition table）了，這也是對初學者最困難的部份，常常因為使用者不知道怎樣進行適當的分割而造成資料的損毀，因此讀者在這裡可要耐心的研讀！

我們首先把硬碟想像成是一間很大的辦公室，如圖 2.5 所示，門口部份有一個櫃子讓你放辦公室相關的訊息。那個櫃子就叫作主要開機磁區（MBR, Master Boot Recorder），而櫃子裡面有兩個抽屜，一個是放置開機管理程式（boot loader），另外一個就是放置磁碟分割表（partition table），這個磁碟分割表就是紀錄整個辦公室隔間隔局的地方。

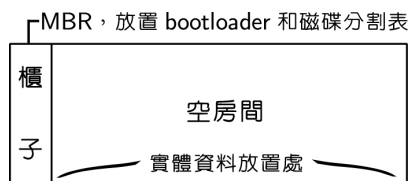


圖 2.5 空硬碟架構示意圖

因為磁碟分割表的空間有限，所以我們只能儲存最多四間隔間的資訊，若你沒有在這四個隔間把整個坪數劃分完，那麼多出來的空間就無法使用唷！而整間辦公室能隔成的房間總共有兩種，一種是主分割區（Primary），另外一種是延伸分割區（Extended），且規定延伸分割區最多只能有一間。那麼延伸分割區是要拿來做什麼用的呢？其實延伸分割區是不能直接使用的，可以看成它是另外一種房間，前面也有櫃子儲存延伸分割區底下的資訊，但是因為裡面沒有傢俱，所以沒辦法直接使用。所以延伸分割區還要再分割成邏輯分割區（Logical），從延伸分割區再分割的邏輯分割區最多可以分割到 12 個，故最後可用的最多分割區數會有 3 個主分割區 + 12 個邏輯分割區 = 15 個分割區，夠用了吧！

表 2.4 不同種類的分割區（以 hda 的 IDE 硬碟為例）

類別	功用
主分割區	可以直接格式化儲存檔案，和延伸分割區加起來的總數只能有 4 個！
延伸分割區	不能直接儲存檔案，裡面必須還要有邏輯分割區來作分割，並且最多只能一個延伸分割區喔！
邏輯分割區	可以直接格式化儲存檔案，是依賴在延伸分割區底下。磁碟代號在 hda5 以後，總共可以有 12 個。

每個分割區在 GNU/Linux 都有它的裝置檔案代號，它的代號就是硬碟裝置代號後面加上數字。例如我們把 4 個分割區都分割成主要分割區，那麼就會如圖 2.6 所示，給的分割區代號分別是 hda1 到 hda4。但是這個時候如果想要超過 4 個分割區呢？那就得要使用延伸分割區啦！我們可以把圖 2.6 最後一個主分割區換成延伸分割區，再把那個延伸分割區分割成 3 個邏輯分割區，最後會如圖 2.7，有 6 個分割區可以用呢！

記得喔！邏輯分割區一定是從 hda5 開始，而前面剛好有 hda1 到 hda4 這 4 個代號，就是給主分割區和延伸分割區瓜分的喔，至於延伸分割區是不能直接用的，最多也只有一個，要延伸分割區裡面的邏輯分割區才可以直接被使用哩！

M B R	hda1	hda2	hda3	hda4
	主	主	主	主

圖 2.6 四個主分割區示意圖

M B R	hda1	hda2	hda3	hda4 延		
	主	主	主	hda5	hda6	hda7
				邏輯	邏輯	邏輯

圖 2.7 三個主分割區加一個延伸分割區示意圖

當然我們也可以用一個主分割區配上一個延伸分割區，如圖 2.8 來達到和圖 2.7 一樣有 6 個分割區的效果，但是我們實務上比較喜歡先把主要分割區用完，再來搭配延伸分割區使用。不管怎樣，只要記得一個規則，主要分割區加延伸分割區最多只能有 4 個，其他的都隨讀者高興來發揮啦！

M B R	hda1	hda2 延伸分割區				
		hda5	hda6	hda7	hda8	hda9
	主	邏輯	邏輯	邏輯	邏輯	邏輯

圖 2.8 一個主分割區加一個延伸分割區示意圖

在 Linux 中一定要被獨立劃分出來的分割區有 / (root) 和 swap，若你只要這樣基本的分割法，那麼安裝的時候選自動分割就是採用這種作法，但是對於有特殊

需求的人，這樣大概不能讓你滿足吧！其他手動規劃方式，筆者將會在 2.3.5 節詳細討論和規劃。

- **/ (root) 分割區**：所有的目錄都是根目錄的子目錄，所以這個分割區一定要有！
- **swap 分割區**：swap 分割區是做什麼用的呢？其實它可大可小，尤其是在記憶體大家都裝 768MB 或 1GB 的年代。它的功能類似微軟 Windows XP 的虛擬記憶體，不過在 Windows XP 裡面虛擬記憶體是單獨一個檔案存放在 C 槽，但是在 GNU/Linux 內則是獨立成為一個分割區。通常 swap 是在跑大程式時候，若需要用到的記憶體比系統擁有的主記憶體還大時，才會先把一些不常用的資料放到虛擬記憶體裡面，然後等要用到時，再把它叫出來。而依照一般舊制經驗法則，swap 是主記憶體的兩倍，但是筆者建議是和主記憶體一樣大就好或者乾脆 512 MB 的 swap 就很夠用啦！

2.3.4 檔案系統格式簡介

在 Microsoft Windows 的世界，硬碟可以格式化成 NTFS、FAT32、FAT16 等等不同的格式。同樣地，在 GNU/Linux 底下也是有很多不同的檔案系統格式可供選擇。目前在 GNU/Linux 底下，比較常用的有這幾種格式：Ext2/Ext3、ReiserFS、XFS 和 JFS 等數種。當然各種格式都有其優缺點，所以我們將做一些簡單的介紹。

除了 Ext2 以外，其他幾種都是日誌型檔案系統。那什麼是日誌型檔案系統呢？就是系統會多用一些額外的空間紀錄硬碟的資料狀態，因而在不正常開關機後，不需整個硬碟重新掃描來恢復正常的系統狀態。

- **Ext2**：此為一非常老舊且不支援日誌系統的檔案系統格式，早期的 Linux 玩家應該還記得吧，在每次不正常關機後，重新開機時錯誤檢查會需要很久，而且在不正常關機下，常常會讓你一次不見很多檔案，現在已經很少人使用這類檔案系統了！
- **Ext3**：為 Ext2 個改良版，所以 Ext2 可以直接升級成為 Ext3 而不必重新格式化，這也可以讓舊的 Ext2 系統更加穩定。而主要和 Ext2 的差別是增加了日誌系統 (metadata)，所以在不正常開關機後，可以迅速使系統恢復。而因為它與舊有的檔案系統相容，因此很多發行版都預設使用 Ext3。但是在實際測試上，它的硬碟使用率其實不佳，大概只有真正空間的 93% 會被

使用到，至於其他效能測試表現則為中等。在格式化與建立檔案系統的時間也是其他檔案系統的數十倍。

- **ReiserFS** : <http://www.namesys.com> : 採用日誌型的檔案系統，為 Hans Reiser 所創，因此以他的名字來命名。技術上使用的是 B*-tree 為基礎的檔案系統，其特色為從處理大型檔案到眾多小檔案都可以用很高的效率處理。實務上 ReiserFS 在處理檔案小於 1k 的小檔案時，效率甚至可以比 Ext3 快約 10 倍，所以 ReiserFS 專長是在處理很多小檔案。而在一般操作，它的效能表現也有中上的程度。
- **XFS** : <http://oss.sgi.com/projects/xfs/> : 繪圖工作站公司 SGI 為了高級繪圖處理器系統 IRIX 所設計的檔案格式，也是屬於日誌型檔案系統。而 SGI 亦將其移植到 GNU/Linux 上。因為原本是針對高效能繪圖設計，且為高階工作站所使用，所以在穩定性和效率上是無庸置疑的。論其在實務上的表現，它在處理各種檔案大小混合的情況下效率最好，並且在一般使用上有不錯的表現。
- **JFS** : <http://jfs.sourceforge.net> : 為全球最大電腦供應商 IBM 為 AIX 系列設計的日誌型檔案系統，技術上使用的是 B+-tree 為基礎的檔案系統和 ReiserFS 使用 B*-tree 不同。IBM AIX 伺服器在很多金融機構上使用，所以穩定性是沒話說的。而它最重要的特色是在處理檔案 I/O 的時候是所有檔案系統裡面最不佔 CPU 資源的，也就是 CPU 使用率最低。而且在這樣節省使用 CPU 的情況下，它的效率表現還有中上以上的程度。

雖然 Ext3 效能不好（在日誌型檔案系統中效率上算是最糟糕的了），那為何還有那麼多人使用？那是因為當時 Ext3 可以直接從 Ext2 升級，而不需要先備份資料，然後格式化後再把檔案複製回去，所以使用人數最多。但這也不能全然怪它，因為它為了和 Ext2 相容，所以背負了很多的歷史包袱。因此我建議新的電腦考慮使用 ReiserFS、XFS 或 JFS。若是以效能為考慮，則可以選擇 ReiserFS 或 XFS。若是系統資源不多，要使用最低的 CPU 使用率，那麼可以選擇 JFS，因為它有著最好的效能資源比。

在網路上有一些檔案系統效能評估，這裡列出來給讀者參考一下：

- <http://www.debian-administration.org/articles/388>
- <http://fsbench.netnation.com/>

- <http://linuxgazette.net/122/TWDT.html>
- <http://linuxgazette.net/102/piszcz.html>

2.3.5 開始規劃你的硬碟

隨著任務的不同，你可以有各種不同的硬碟規劃。筆者假設你使用 80 GB 的 IDE 硬碟，記憶體 512 MB，因此規劃 swap 是 512 MB，硬碟裝置代號是 hda。

筆者提供以下幾種不同的硬碟規劃，分別是適合入門，進階和伺服器所需要，讀者可以視需求自行參考。不過你在進行分割之後，有可能會因為換算的關係，少個幾 MB，那是沒有關係的，只要不要太離譜，都應該是沒問題的。

表 2.5 新手入門桌面分割區參考

掛載點	裝置	說明	容量
/	/dev/hda1 (主分割區)	建議採用 ReiserFS	79.5GB
swap	/dev/hda2 (主分割區)	虛擬記憶體	512MB

表 2.6 進階桌面分割區參考

掛載點	裝置	說明	容量
/	/dev/hda1 (主分割區)	建議採用 ReiserFS	15GB
/home	/dev/hda2 (主分割區)	把/home (家目錄) 獨立為另外一個分割區，若是系統重灌時，個人資料就可以留下來了。建議採用 ReiserFS 格式	64GB
/boot	/dev/hda3 (主分割區)	放置開機相關的核心檔案，建議採用 Ext3 或 ReiserFS	128MB
swap	/dev/hda4 (主分割區)	虛擬記憶體	512MB

表 2.7 伺服器分割區參考

掛載點	裝置	說明	容量
/	/dev/hda1 (主分割區)	建議採用 ReiserFS	15GB
/home	/dev/hda2 (主分割區)	把/home (家目錄) 獨立為另外一個分割區,若是系統重灌時,個人資料就可以留下來了。格式上建議採用 ReiserFS	64GB
/boot	/dev/hda3 (主分割區)	放置開機相關的核心檔案,建議採用 Ext3 或 ReiserFS	128MB
	/dev/hda4 (延伸分割區)	不能直接使用,需要再細分成下面的邏輯分割區	18GB
/var/log	/dev/hda5 (邏輯分割區)	放置一些系統紀錄檔,建議最好是獨立出來。格式上建議採用 ReiserFS。	2GB
/var/spool	/dev/hda6 (邏輯分割區)	可能是 E-Mail 等佇列存放的地方,建議最好是獨立出來。格式上建議採用 ReiserFS。	15GB
swap	/dev/hda7 (邏輯分割區)	虛擬記憶體。	1GB

每個分割區都是可以存放獨立的檔案系統, 互相不影響, 所以讀者也可以在 hda1 安裝 Windows XP, 然後 hda2 安裝 GNU/Linux, 再透過 MBR 上的 bootloder 來選擇要用哪個分割區來開機。如果讀者真的想要使用雙系統的話, 建議是先安裝 Windows XP, 並且在安裝時留下剩下的空間來做額外的分割區給 Ubuntu 使用, 如此在安裝 Ubuntu 以後, Ubuntu 的 bootloder—GRUB 將會自動設定成開機的時候可以有雙系統供你選擇。請記得最好不要先安裝 Ubuntu 再安裝 Windows XP, 這樣 GRUB 的設定會被 Windows XP 洗掉, 那你就得要手動重新安裝 GRUB 到 MBR 了!

若是你真的不想重灌 Windows XP, 可以先重整硬碟, 然後用 SPFDISK 這個工具程式把硬碟的分割表變更大小, 這樣就可以擠出空間來裝 Ubuntu。但這個方法特別留意, 並不一定會成功, 而且也許會造成你的資料遺失。

表 2.8 Ubuntu 和 Windows XP 雙系統分割區參考

掛載點	裝置	說明	容量
XP 安裝 在 C 槽	/dev/hda1 (主分割區)	建議先安裝 Windows XP	30GB
/	/dev/hda2 (主分割區)	建議採用 ReiserFS	49GB
swap	/dev/hda3 (主分割區)	虛擬記憶體	512MB

CHAPTER 3

開始安裝 Ubuntu

3.1 用光碟開機

只要是使用可開機光碟片安裝任何作業系統，一開始所做的第一件事，就是設定 BIOS 讓你的電腦從光碟開機，不過這個部份比較不適合在此講解，因為各廠牌主機板的 BIOS 設定方式都不一樣，如果沒自己灌過作業系統的讀者別擔心，因為很簡單。基本上你買回來的主機板都會有說明書，即使說明書不見了也沒關係，網路上也有很多教學，只要知道了你的主機版廠牌型號，用 google search 一下都會很多教學的文章。

3.2 多重開機的安裝技巧

顧名思義，就是在一部電腦上安裝多個作業系統。如果你是 Linux 新手，那你一定會有這樣的問題，多重作業系統要怎麼安裝呢？要怎麼選擇作業系統啊？在這個小節筆者就跟你講解一下。首先，在安裝時不管你是要安裝什麼系統，要記得 Linux 是最慢裝上去的，因為 Linux 幾乎可以支援所有的系統檔案格式。假如讀者想要在一台電腦上裝 Linux 和 Windows，那麼不管你是用一顆硬碟或是多顆，只要用 spfdisk 或是 Windows 正版光碟安裝時，預留一下安裝 Linux 系統的空間即可。接著在你看過了上面第二章，了解過了 Linux 是怎麼對硬碟與分割區

做編號時，那麼你應該可以分辨出來，剛才裝上去的 Windows 是哪一個硬碟與分割區，然後只要在安裝 Ubuntu 時，在設定磁碟分割作業時將 Windows 磁區掛載上去，就可以在安裝完 Ubuntu 後在桌面上看到 Windows 的磁區囉！讀者可以修改一下掛載點，到時候會比較好分辨。例如：在下面安裝時，掛載點可以設成 /media/D 這樣子來代表 D 槽。要讀取 Windows 的 NTFS 磁區，可以閱讀第十三章的說明。

3.3 開始安裝

本書上附的光碟是 DVD 的 386 版本，若讀書想要其他版本的 Ubuntu 光碟，其參考第二章。這裡筆者就簡單介紹一下，Live CD 安裝與文字介面模式安裝，那麼接下來就是放入光碟開機囉。第一個畫面有八個選項。可以選擇第一項，啟動 Ubuntu，這個選項很好用，是用 Live CD 的方式開啟 Ubuntu，當然也可以直接以這樣的方式，安裝 Ubuntu。

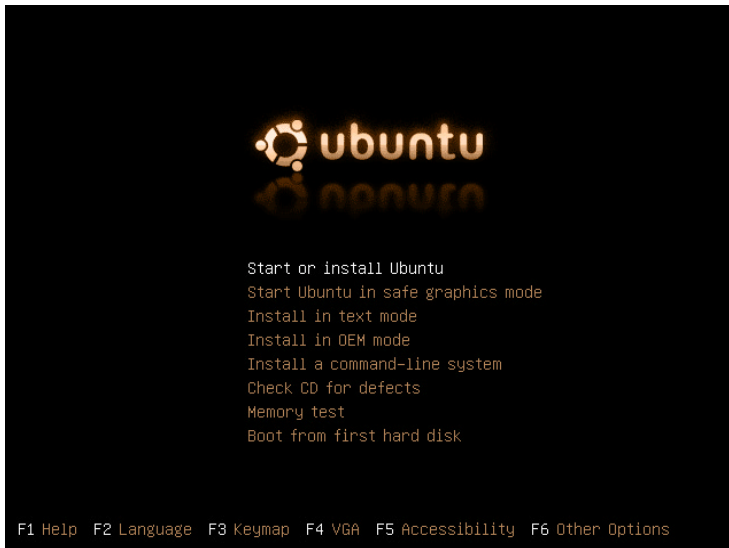


圖 3.1 Ubuntu 開機選單

先按下 F2 選擇中文(繁體)



圖 3.2 中文(繁體)

3.3.1 Live CD 模式安裝 Ubuntu

光碟放入後選擇第一項，啟動 Ubuntu。



圖 3.3 以 Live CD 模式安裝

開機過程的圖就不附上囉！然後直接會進到桌面，讀者也可以先試玩一下 Ubuntu。接著請讀者按下桌面上的 Install 開始安裝。

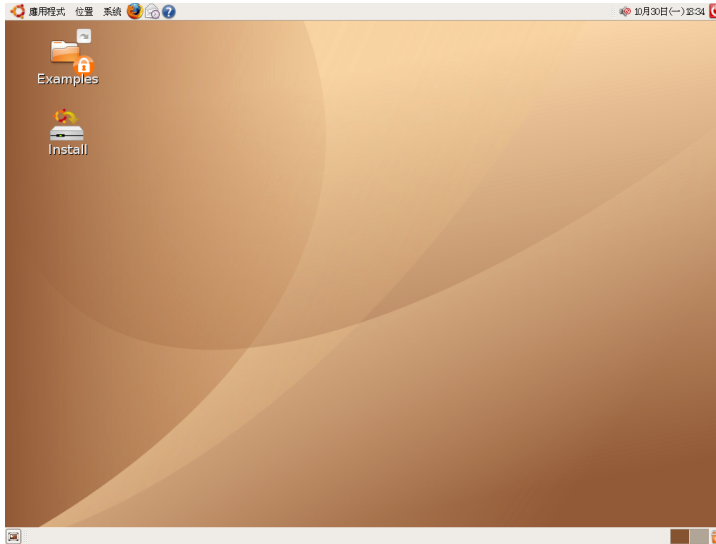


圖 3.4 Ubuntu 桌面

接著選擇中文(繁體)的選項。

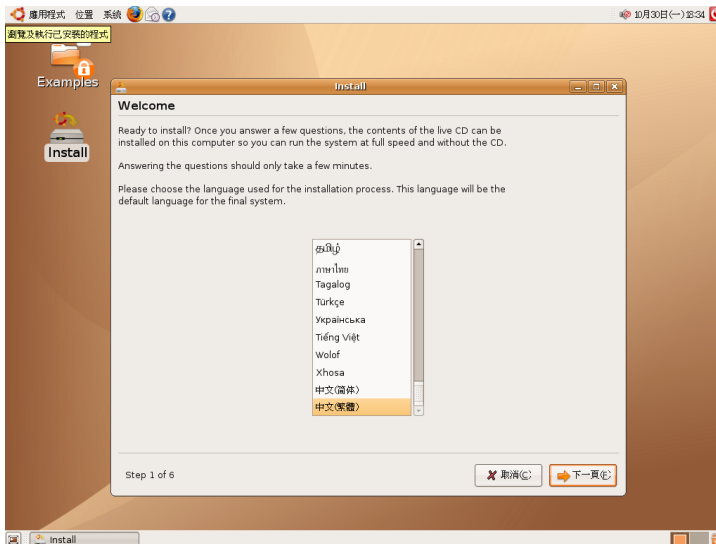


圖 3.5 選擇繁體中文

然後選擇你的時區，如果你前面已經選擇繁體中文，那麼系統會自動幫你設成 Taipei GMT+8:00，當然囉！讀者可以再按下**設定時間**來自行調整哦。

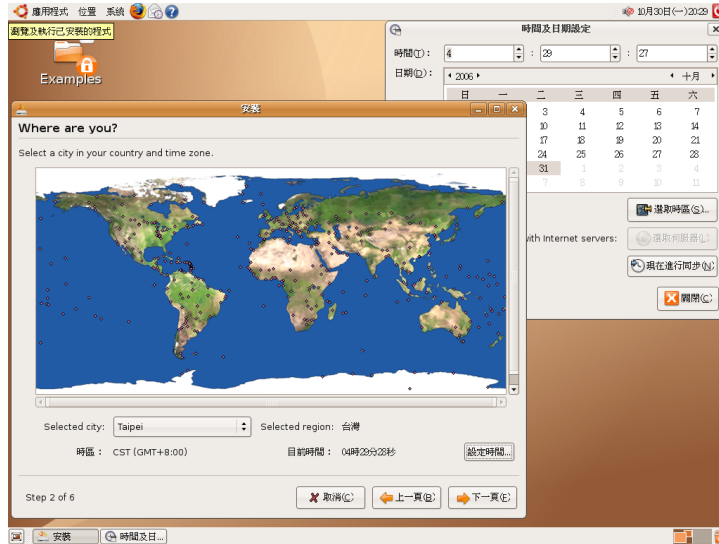


圖 3.6 設定時區與時間

選擇你的鍵盤排列方式。通常台灣都是用美語的 QWERTY 的排列，所以這裡就選 U. S. English。

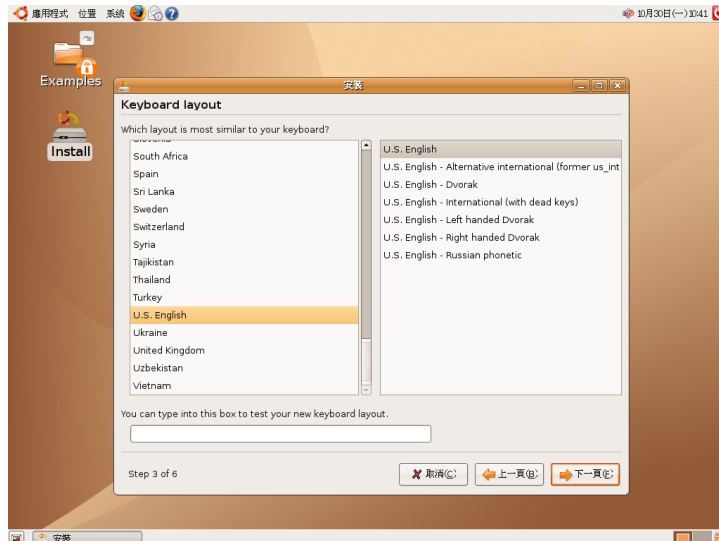


圖 3.7 選擇你的鍵盤排列方式

接下來輸入一下，姓名、帳號、密碼以及你的電腦名稱。



圖 3.8 輸入基本資訊

磁碟分割，選擇第一個刪除整顆磁碟的話，請確定你的這顆磁碟是完全都要給 Ubuntu 用的，並且用這個選項的話，系統只會自動幫你割兩個磁區，即根目錄「/」與 SWAP。如果讀者不知道要怎麼分割的話，可以先用這個選項。不過也可以照著下面的步驟來分割，選擇第二個選項，手動編輯磁碟分割表。以下範例筆者是用 Vmware 弄了 4G 的空間出來，是因為要抓安裝的畫面才這樣做的，不過事實上其實你沒有安裝很多程式的話，大約 4~6 GB 的空間大概就夠用了。

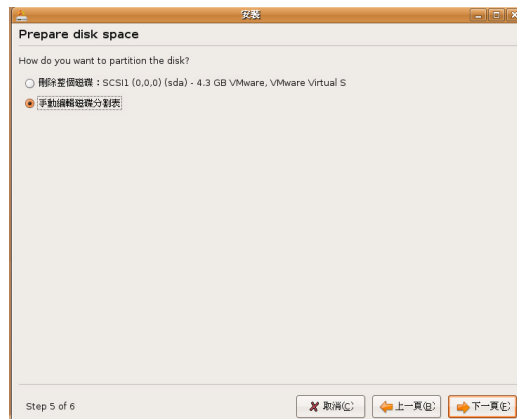


圖 3.9 手動編輯磁碟分割表

接著你可以看到有一顆 4G 的硬碟，上面標示著沒有配置。或者是你先前已經有安裝了 Windows，那麼在這裡可以看到已經配置的 Windows 磁區，如果是這樣的話這裡就先不要動到，等待會再掛載上去即可。接下來點選沒有配置的空間，然後點選上面的新增。接著會出現新增磁碟標籤的視窗，選擇建立（注意：這個動作會讓你磁碟上的資料都不見，所以在做這動作前，請先將你硬碟裡的資料備份好哦！）。在這裡要說明一下，讀者可以看到圖上方裝置是 /dev/sda，那是因為用 VMware 的關係。如果是用一般的硬碟應該會是 /dev/hda、/dev/hdb 等等的。

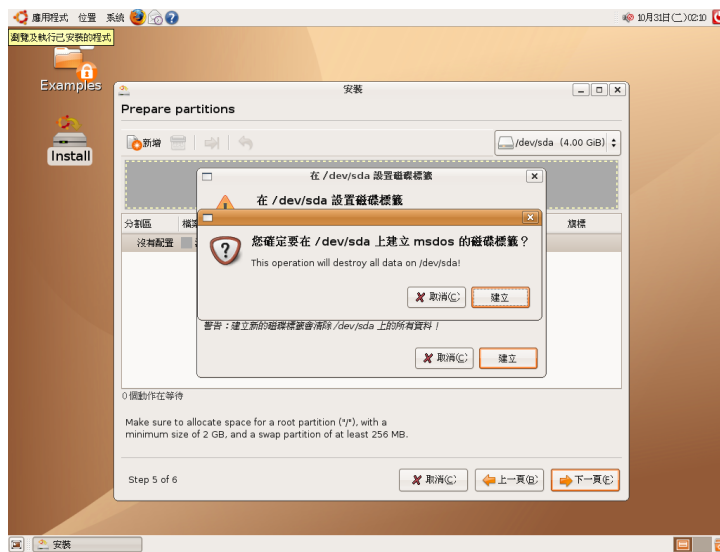


圖 3.10 新增磁碟標籤

接著會出現新增分割區的視窗，可以在 New Size 的地方輸入你要的大小，以及選擇檔案系統，讀者也可以參考上個章節來做分割。那麼因為是示範，所以在這裡就只切成三個磁區，在這裡先輸入 3000，也就是大概 3GB（事實上應該比較小），來當做「/」根目錄。

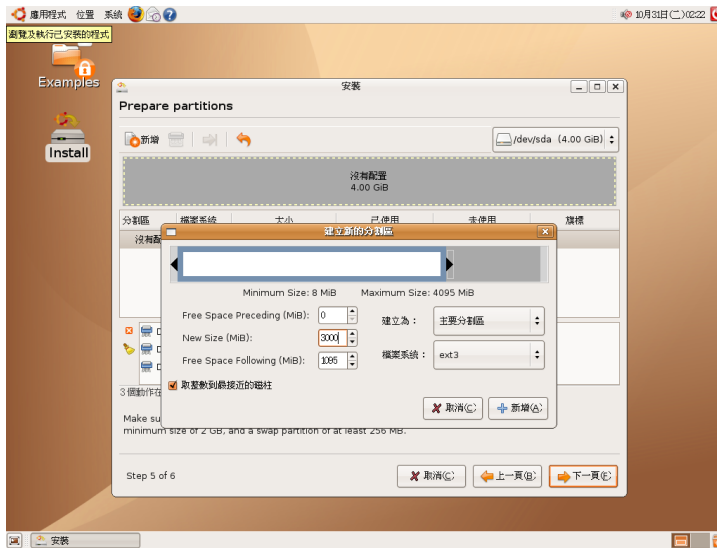


圖 3.11 新增根目錄

接下來一樣，在沒有配置的地方按右鍵選**新增**，這裡輸入 700 MB 當做是「/home」。

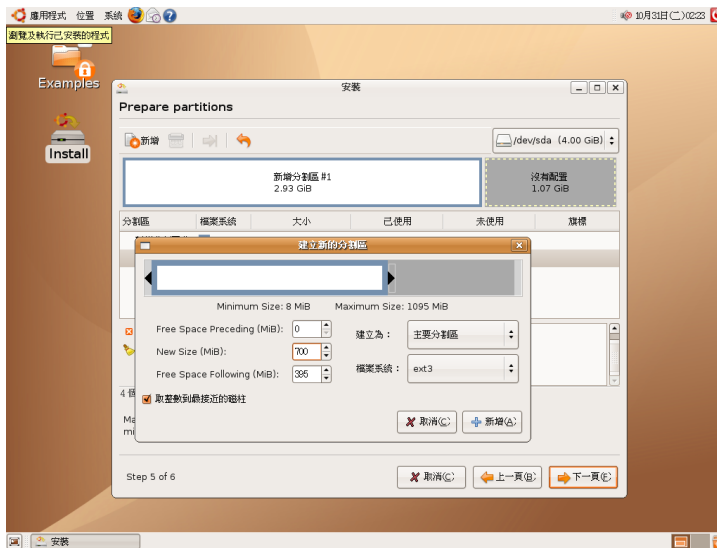


圖 3.12 新增家目錄

然後剩下的三百多 MB 留給 swap 用，請記得在檔案系統那選 linux-swaps 哦。

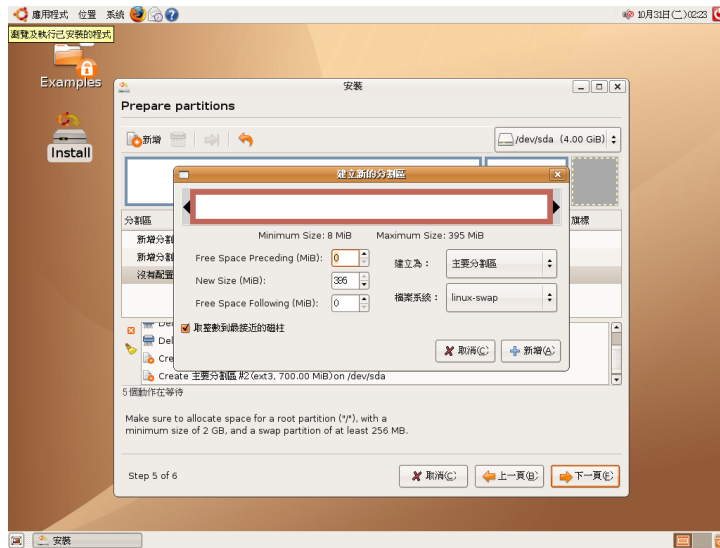


圖 3.13 新增置換空間

確認一下是不是跟剛才分割的一樣，是的話就按下一頁。筆者在這裡再次強調，這個分割方式是屬於測試用的，要請讀者記得，自己分割的時候，「/」根目錄可以再大一點，而家目錄則是能給多少就給多少，因為家目錄是用來放自己的資料用的。以免分割區太小到時候檔案放不下。

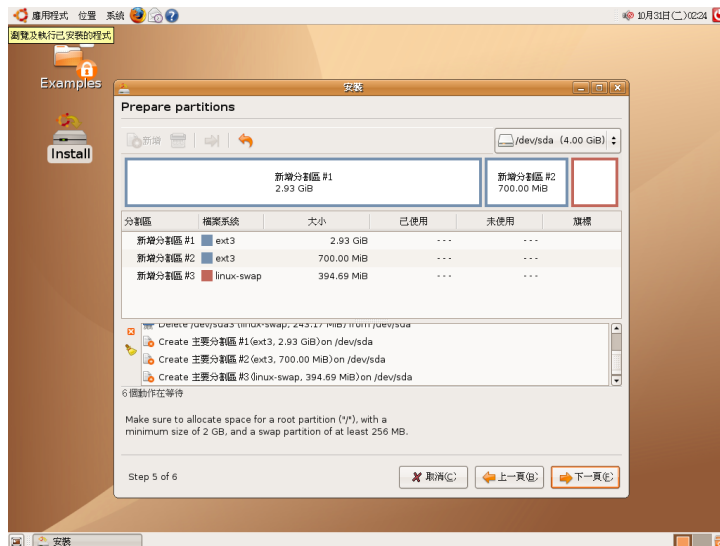


圖 3.14 確認一下分割的大小

接著將分割好的資訊寫到硬碟裡，按下**套用**。

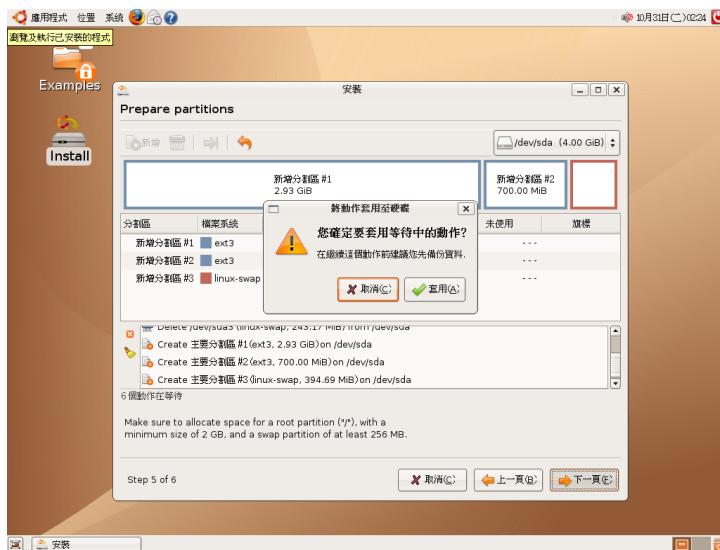


圖 3.15 套用分割

套用完了以後按**關閉**。

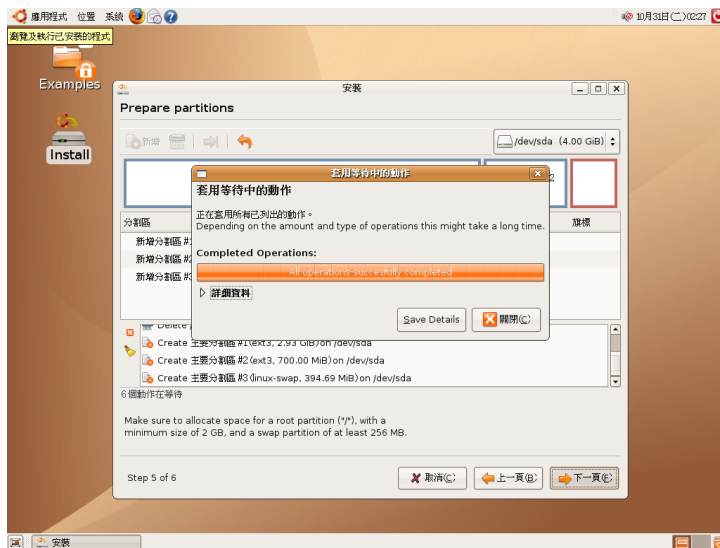


圖 3.16 完成磁碟分割套用

然後會出現選擇掛載點，在這裡我們剛才割好的 Partition（磁區），不見得都會自動出現，要自己按出來，讀者可以按照剛才分割的順序，將 Partition 按一下旁邊的箭頭，選出來，以及你安裝的 Windows 的磁區一樣選出來。然後依照剛才分割的大小，在左邊輸入掛載點。Windows 磁區也是一樣，例如 Windows D 槽，可以輸入 /media/D，這樣在你安裝好 Ubuntu 後，就可以在桌面上，或是儲存媒體裡使用 Windows 磁區。

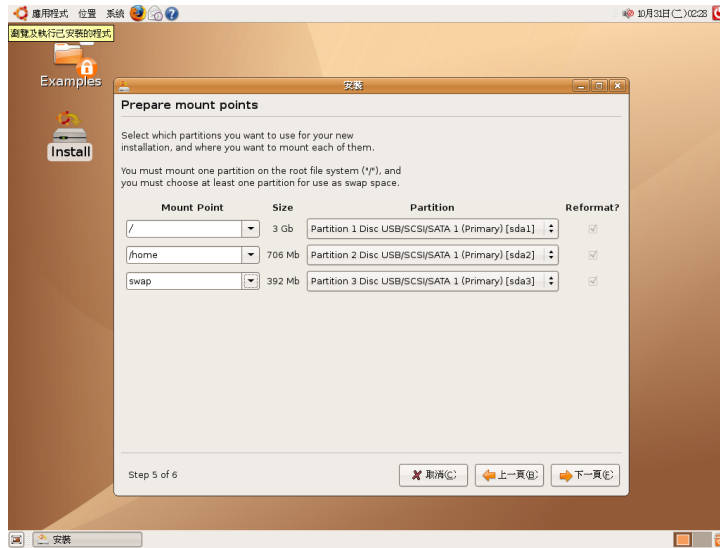


圖 3.17 選擇掛載點

接著就可以按下 Install 開始安裝囉！

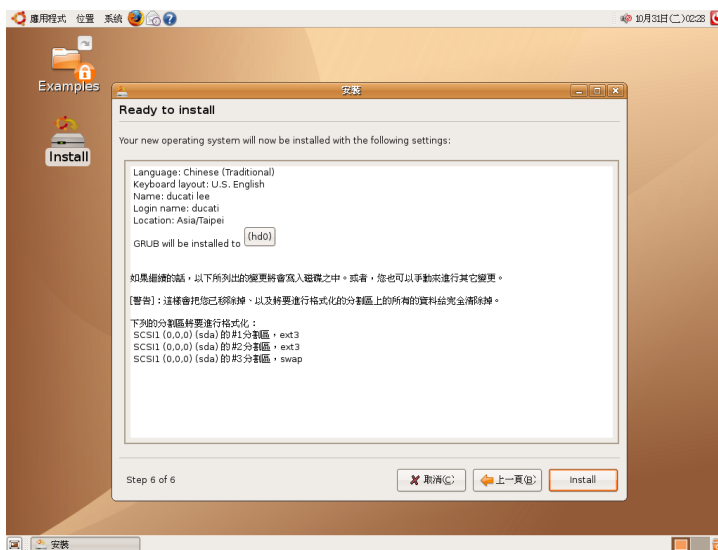


圖 3.18 開始安裝

然後經過漫長的等待，你就可以按下 **Restart Now** 取出光碟，並開機到你專屬的 Ubuntu 裡囉。

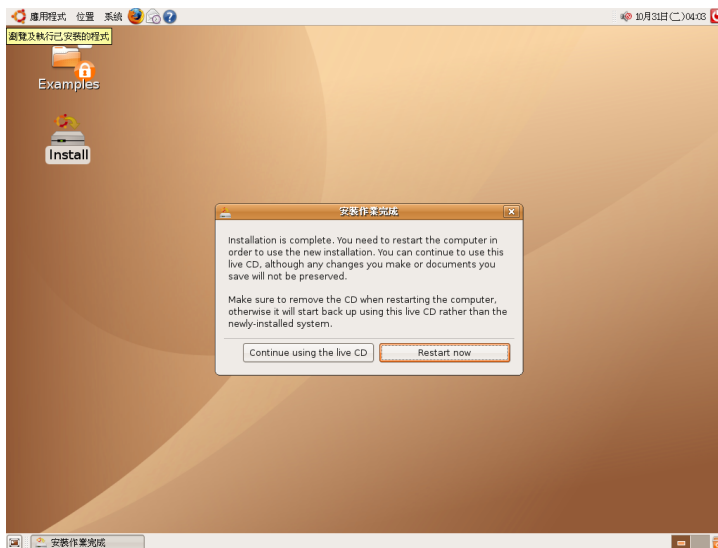


圖 3.19 重新開機

3.3.2 文字模式安裝 Ubuntu

要啟用這個模式，在用 DVD 版光碟開機並選擇中文（繁體）後，選擇安裝到硬碟的選項。



圖 3.20 選擇安裝到硬碟

接著會問你是否要偵測鍵盤，在這裡選否，我們自己選就好了。



圖 3.21 偵測鍵盤

選擇 U. S. English，因為通常台灣鍵盤都是 QWERTY，的排列方式。第二頁一樣也是選 U. S. English。



圖 3.22 選擇鍵盤排列

接下來會開始偵測硬體，掃描光碟機、載入額外的元件，偵測網路硬體、用 DHCP 設定網路（如果你有 IP 分享器的話，在這裡會自動抓到 IP）。這裡就不附圖囉！因為沒有選項。然後輸入你的主機名稱。

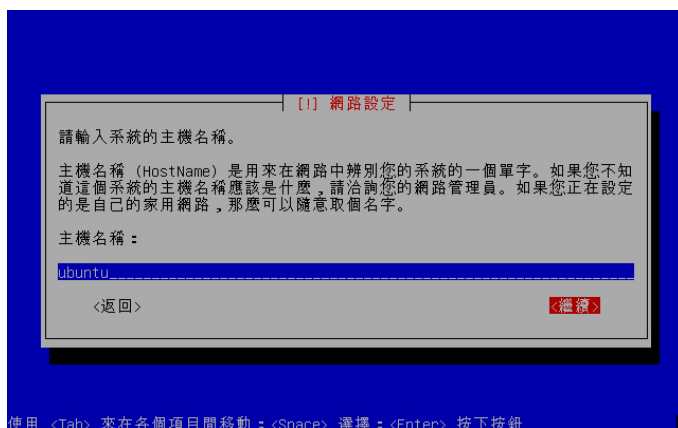


圖 3.23 輸入主機名稱

接著會出現，分割磁碟作業的視窗。在這裡也一樣哦！要參考第二章來分割，底下的圖是測試用的，所以不建議讀者照著下面的大小分割。選擇第一個的話，

會刪除整個磁碟，包括 Windows 磁區的部份，不過同時也適合懶人來用。但是在這裡我們選擇手動編輯。

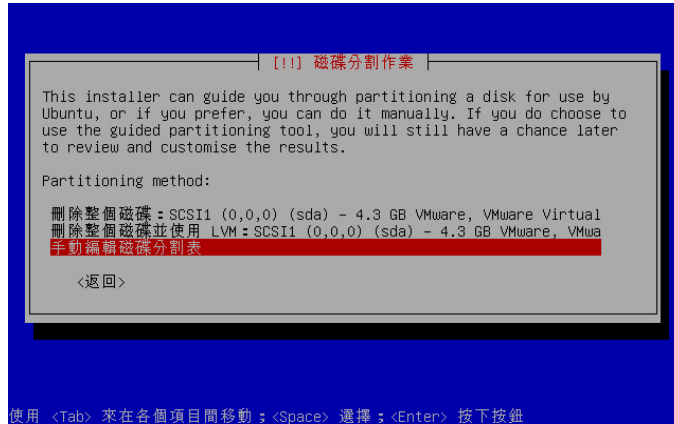


圖 3.24 選擇手動編輯

然後呢，我們可以看到，筆者先前用 Vmware 建好的 4GB 的磁碟空間。選擇並按 Enter（可以用上下鍵及 Tab 切換選項）。

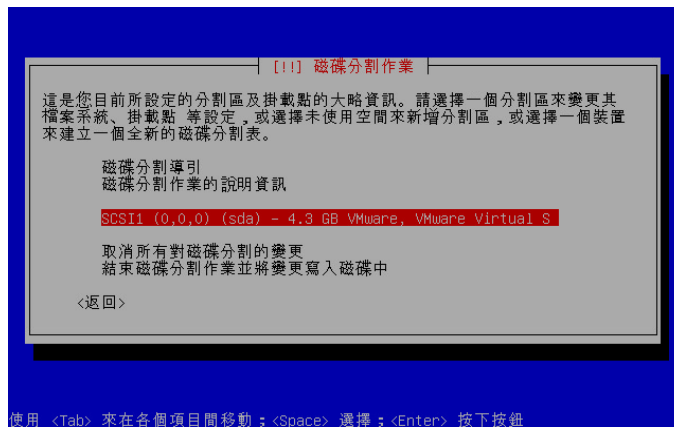


圖 3.25 選擇要分割的硬碟

建立全新的磁碟分割表，選是。

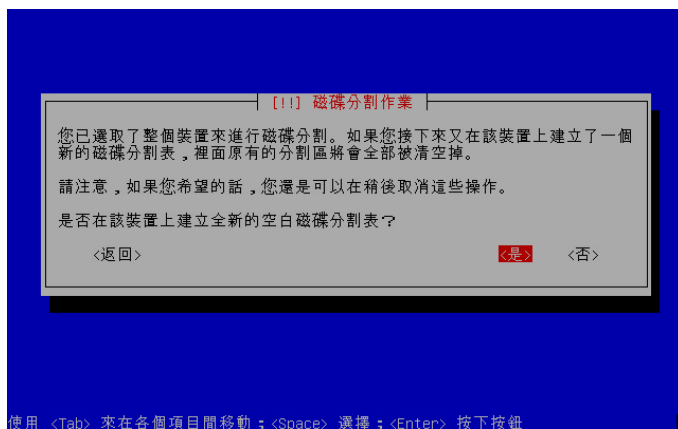


圖 3.26 建立全新的磁碟分割表

接下來就可以看到一個空的磁區，按下 Enter 來建立新的分割區。

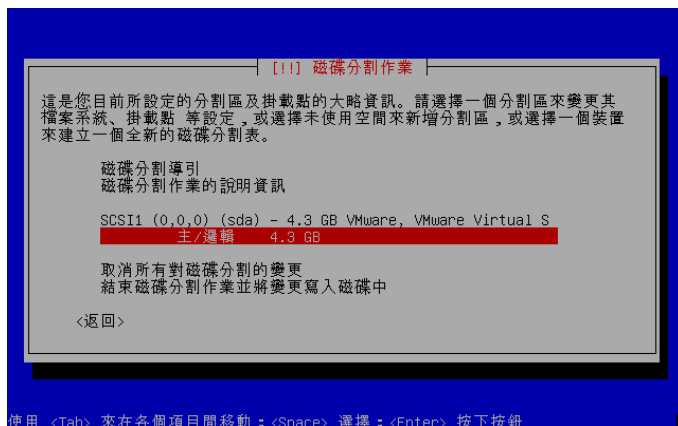


圖 3.27 選擇空的磁區

選擇建立新分割區。

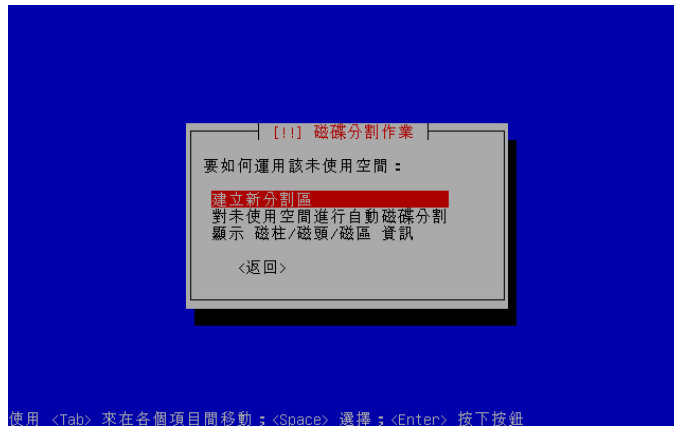


圖 3.28 建立新分割區

配置容量，請讀者自行辨斷哦！最好不要跟我一樣，因為這是測試用的。通常第一個分割區「/」，筆者都是給 6 GB。或是第一個分割區，筆者會設 /boot 30 MB，/home 可以給多大就多大。分割的時候請注意要看你的電腦用途，以及硬碟大小來配額。

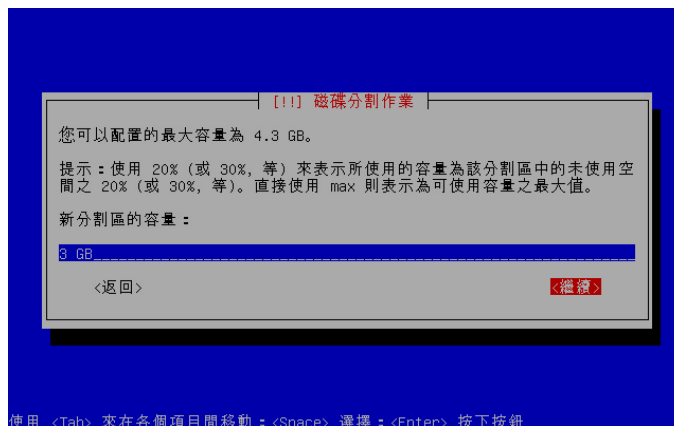


圖 3.29 配置根目錄容量

建立在主分割區。



圖 3.30 建立在主分割區

選擇新分割區在開始的位置。

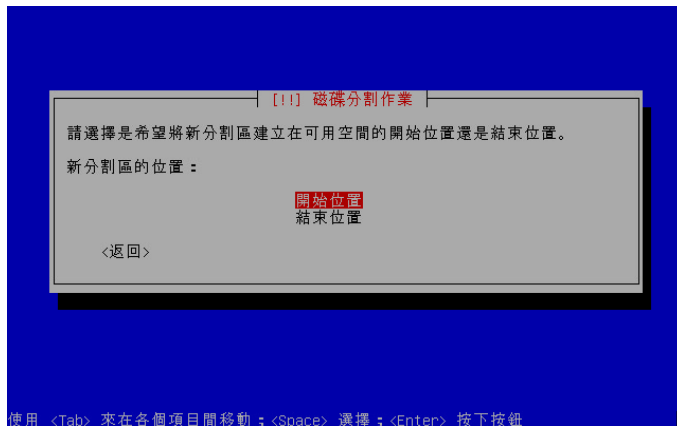


圖 3.31 開始位置

接著這裡是很重要的部份，第一個欄位是用來選擇檔案系統，讀者可以選擇 ext3、ext2、reiserfs、swap 等等。

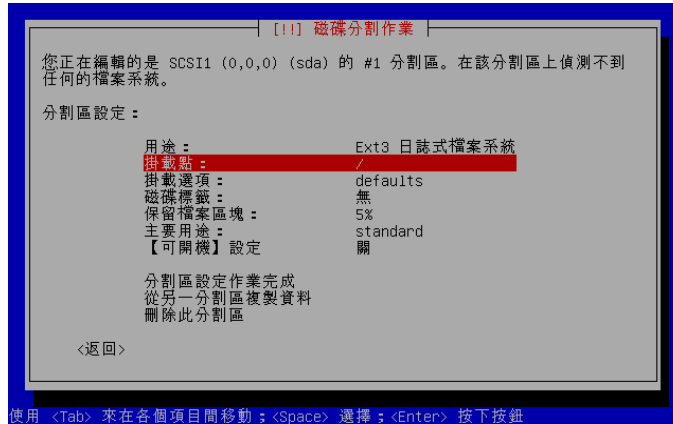


圖 3.32 磁碟分割作業

第二個欄位是選擇掛載點。

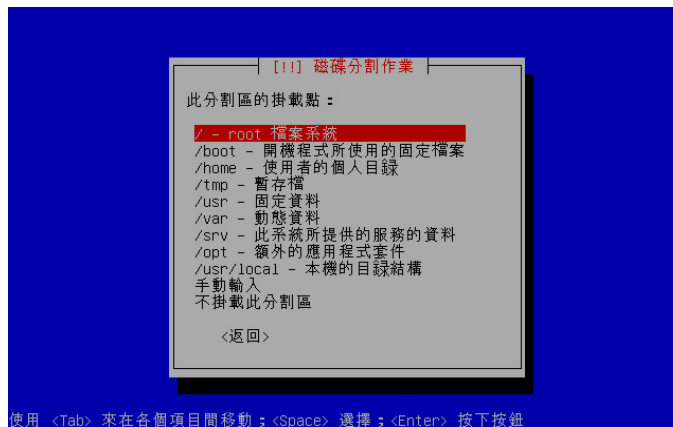


圖 3.33 選擇掛載點

這樣這個磁區就分割好了哦！其他的磁區與掛載點，也是這樣子設就可以了。先分割好大小後，再選擇檔案格式與掛載點。不管是任何檔案系統與掛載點，都是這樣設定。而 swap 的設定方式，給完 swap 大小後，只要在用途的地方，**選擇置換空間**即可。設定好後按下**分割區設定完成**。

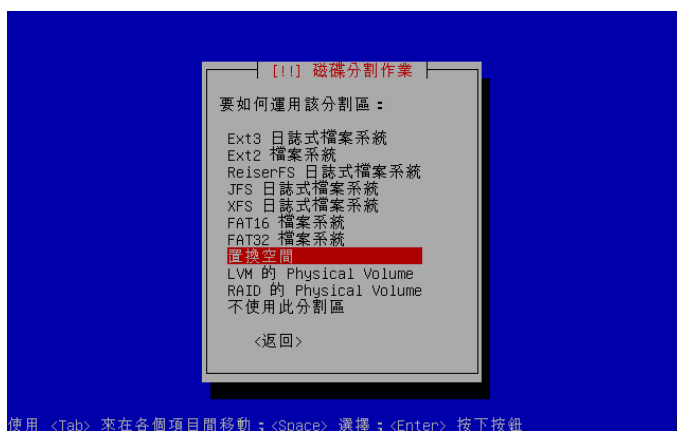


圖 3.34 設定 swap 磁區

所有磁區都設定好了以後，就可以將硬碟分割表的資料，寫入硬碟中。

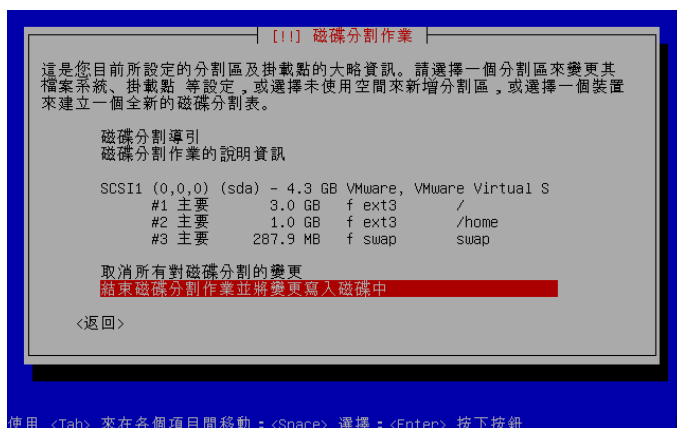


圖 3.35 結束磁碟分割作業

如果你確定分割是對的，那麼就按下是。

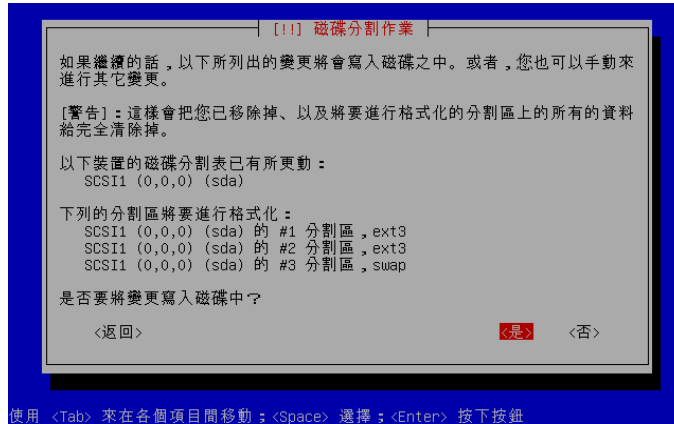


圖 3.36 將分割磁區資料表寫入硬碟

是否將硬體時鐘設定為 UTC，按下是。

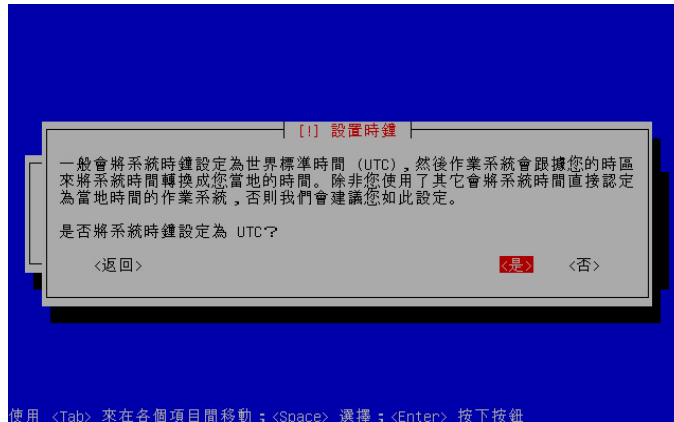


圖 3.37 設定硬體時鐘

輸入你的姓名。

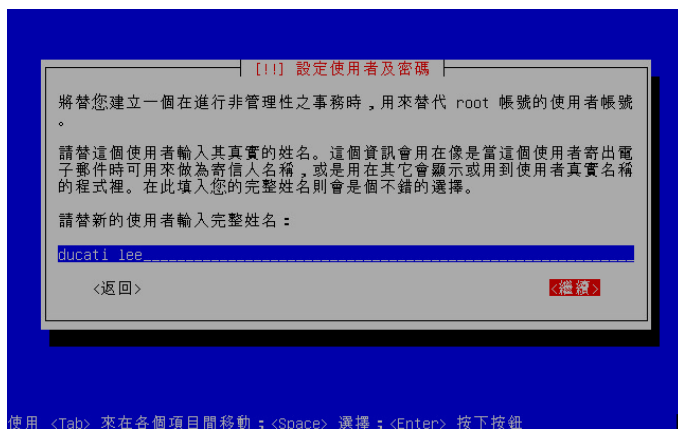


圖 3.38 輸入姓名

輸入你的帳號，這裡要先讓讀者了解一下，在 Ubuntu 安裝過程中，是不會出現 root 這個超級使用者的設定，因為 Ubuntu 內建就已經有 sudo (super do) 這個指令了，所以你現在要設的這個帳號，就有 sudo 權限，這是為了安全性的考量。讀者可以在日後，用 sudo 再去設定 root 的密碼。

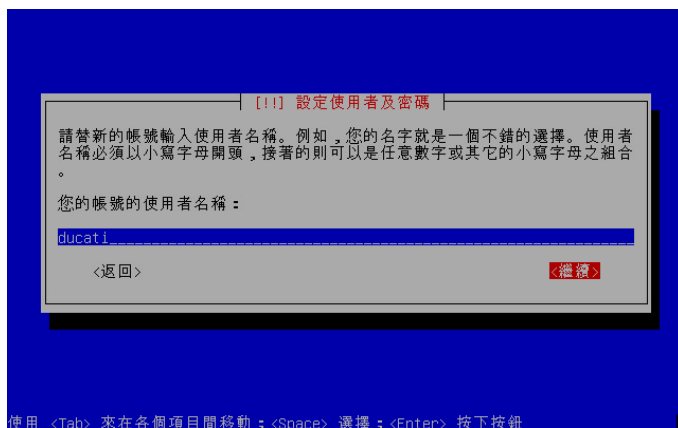


圖 3.39 輸入你想用來登入的帳號

輸入密碼，要兩次哦！

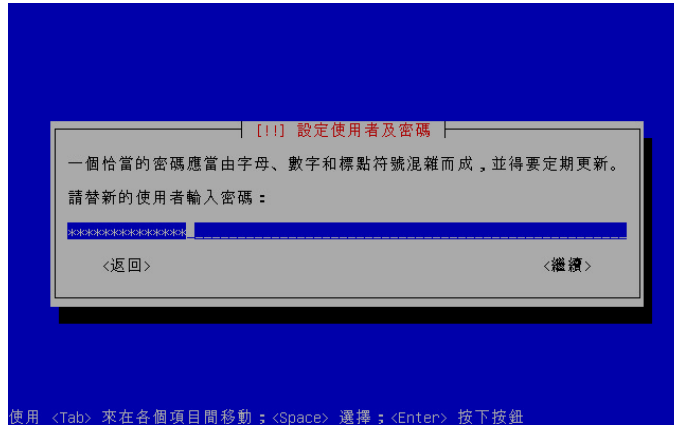


圖 3.40 輸入密碼

接著會出現設定解析度的選項，這就要看讀者的硬體可以設定多高的解析度了。然後就請耐心等待安裝完成囉！



圖 3.41 設定解析度

嘿嘿！安裝完成後就請拿出光碟按下繼續來開機啦！看一下新的開機過程畫面。



圖 3.42 開機過程畫面

CHAPTER 4

Ubuntu & Unix Like 基本知識

在讀者看過了第二章的 GNU/Linux 目錄的架構後，並且在安裝完 Ubuntu 想先試試前，你必需要更加了解 Linux 的文字模式。讀者一定有個疑問？Ubuntu 不是都弄得好好的？為什麼還要學習文字介面呢？雖然只要照著本書指令操作都可以成功，但要是想新增自己想要的軟體、想要從家裡登入到別台電腦，而一切的一切為什麼又要打這樣的指令呢？學習 shell 以及 Linux 的指令到底有什麼好處？文字介面是 Linux 不可不學的，也可以說是非常重要的，如果你想對 Linux 有更深一層的了解，那麼一定要看看這個章節。

在這裡非常感謝晏璿兄 (yenjinc)，他是好幾年前筆者在宇新大哥 (lloyd) 的 Linux 補習班課程中一同認識的好友兼同好。這是他在政大應用數學系 unix-like 暑期研習課程中，帶領學生的入門手冊，他熱情的提供原始文件來友情支援。以下文件由筆者額外編修過。

註解：文件原始的資料可由晏璿兄的網誌中找到：<http://yenjinc.info/wp-content/uploads/2006/10/unix-like-tutorial-951021.pdf>。

4.1 絕對路徑與相對路徑

在 Linux / Unix-like 系統當中，所有的目錄都是掛載於 /（根目錄）底下，通常我們可以使用「cd」這個指令來切換目錄，而切換目錄的方式，有兩種方式，一種是使用「絕對路徑」，另外一種是使用「相對路徑」，關於絕對路徑與相對路徑的說明，筆者在下方來補充。

- **絕對路徑**：凡切換所有路徑都是以「/」作為開頭，則屬於絕對路徑表示法。
- **相對路徑**：凡路徑的表示，均不以「/」作為開頭，則屬於相對路徑表示法。

舉例說明：

「絕對路徑」（凡切換所有路徑都是以「/」作為開頭，則屬於絕對路徑表示法。）

```
cyj@riemann:~$ cd /                (到根目錄)
cyj@riemann:/$ cd /home/cyj        (到 home 底下的 cyj 目錄)
cyj@riemann:~$ cd /etc             (到 etc 目錄)
cyj@riemann:/etc$ cd /usr/share/    (到 usr 底下的 share 目錄)
```

「相對路徑」（凡路徑的表示，均不以「/」作為開頭，則屬於相對路徑表示法。）

```
cyj@riemann:/usr/share$ cd ../../ (從 /usr/share 目錄往上走兩層)
cyj@riemann:/$ cd home/cyj/       (在根目錄中切換到 home 底下的 cyj 目錄)
cyj@riemann:~$ cd ../../usr/share/ (在 /home/cyj 目錄中往上走兩層，進入到 usr 底下的 share 目錄)
```

提到「相對路徑」表示方法，在此有四個符號表示方法，大家一定要熟悉唷！

- 「.」：「點」符號，代表「現行目錄 (current directory)」的意思。
- 「..」：「點點」符號，代表「上一層目錄」的意思。
- 「~」：「波浪」符號，代表「使用者家目錄」的意思。
- 「-」：「-」連結符號，代表「上一個使用目錄」的意思。

下面一樣舉些例子讓各位可以更清楚了解這些路徑符號的意義：

「點 (.)」代表「現行目錄 (current directory)」的意思。

```
cyj@riemann:~$ cd .          (進入現行目錄)
cyj@riemann:~$ pwd          (使用 pwd 指令查看現在所在目錄位置)
/home/cyj                   (一樣在家目錄當中，沒有更換位置)
```

「點點 (..)」代表「上一層目錄」的意思。

```
cyj@riemann:~$ cd ..        (從家目錄，到「上一層」目錄)
cyj@riemann:/home$ cd ..    (於是來到了 home 目錄，接著繼續往「上一層」移動)
cyj@riemann:/$ pwd         (查看現在所在目錄位置)
/                           (由於剛才從 home 又往上爬一層，於是查詢的結果為根目錄)
```

「波浪 (~)」代表「使用者家目錄」的意思。

```
cyj@riemann:/$ cd ~        (從根目錄，回到自己的家目錄)
cyj@riemann:~$ pwd        (查看現在所在目錄位置)
/home/cyj                 (已經回到自己的家目錄了)
cyj@riemann:~$ cd ~cyj    (進入到 cyj 的家目錄中)
cyj@riemann:~$ cd ~yenlung (進入到 yenlung 的家目錄中)
cyj@riemann:/home/yenlung$ cd ( cd 與 cd ~ 是一樣的意思，都是回到自己的家目錄中)
```

「- (-)」代表「上一個使用目錄」的意思。

```
cyj@riemann:~$ cd ~yenlung (進入到 yenlung 的家目錄中)
cyj@riemann:/home/yenlung$ cd ( cd 與 cd ~ 是一樣的意思，都是回到自己的家目錄中)
cyj@riemann:~$ cd -        (讓它再回到剛才上一個使用的目錄位置)
/home/yenlung              (於是又回到了 yenlung 的家目錄裡)
cyj@riemann:/home/yenlung$ cd - (這時候在使用一次 cd - ，會再回到上一個使用的目錄位置)
/home/cyj                  (所以又回到了自己的家目錄)
cyj@riemann:~$ cd -        (再使用一次！又會再回到上一個使用的目錄位置)
/home/yenlung              (於是又回到了 yenlung 的家目錄囉！)
```

這個「cd -」的意思，比較像是「上一頁」的意思！就好比各位在瀏覽網頁時，時常會按下「回上一頁」的按鈕一樣，這裡的 cd - 就是讓你可以從現在的目錄回到「上一個使用的目錄」哦！

OK！以上這些就是基礎的「絕對路徑/相對路徑」的基礎概念啦！筆者建議邊看書時，可以一邊實做來個現學現賣哦！

4.2 遠端登入

如果各位的 Linux 是安裝在外地的其他電腦上，那麼就必須要「遠端登入」才能夠在遠端的主機上進行練習與操作，假使你目前已經在 Linux 或其他 Unix-like 作業系統下，那麼就不需要額外安裝任何軟體，直接使用系統的 ssh 或 telnet 指令就可以直接登入到遠端的機器啦！假使各位是在 Windows 下想要登入到遠端的 Linux 主機，有兩種方式。第一種可以打開「命令提示字元」，使用 Windows 內建的 telnet 命令來登入。第二種就是使用遠端登入軟體「PuTTY」（<http://www.putty.nl/download.html>），或是用 PieTTY（<http://ntu.csie.org/~piaip/pietty/>），PieTTY 的使用方式與 PuTTY 大同小異。

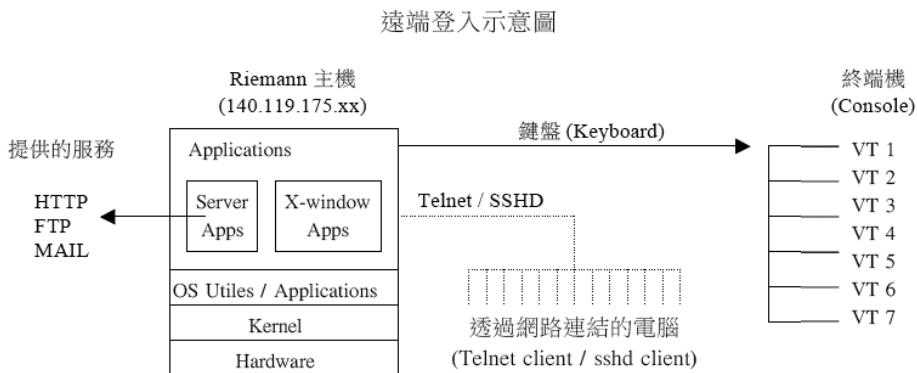


圖 4.1 遠端登入示意圖

在 Linux 系統上主要有兩種登入的方式：

- **第一種**：各位可以親自走到主機的面前，拿起鍵盤，輸入帳號、密碼，這就是一個最基本的登入方法了！一般而言，預設可登入的終端機（Virtual Console）有六個，分別是從 VT1 到 VT6，而 VT7 通常是作為 X-window（圖形化桌面）的登入與使用，如果各位想要切換所使用的虛擬終端機，那麼只要在鍵盤上按下「Ctrl」+「Alt」+「F2」就可以切換到第二個 Virtual Console (tty2)，因此類推「F1」到「F7」就可以讓你從 tty1 切換到 tty7 囉！

- **第二種**：在 Linux / Unix-like 系統上，最讓人津津樂道的一個特點是「你永遠可以不需要在電腦的面前來使用電腦」（有沒有繞口令的感覺 :-D），換句話說，只要透過「網路」，你在「何時何地」，都可以來使用這台電腦。從上面的「遠端登入示意圖」可以看到，在 Linux 主機上，可以透過開啟「Telnet」或者「SSHD」的服務來啟動「遠端登入」的功能，一旦這些服務啟動之後，所有的電腦都可以透過網路直接登入到主機來使用。

4.3 登入與登出

遠端登入到主機時，必須填入帳號跟密碼，以下是一個簡單的示範例子：

```
cyj@X40:~> ssh yenjinc@yenjinc.info (由於筆者在 Linux 環境下，於是直接使用 ssh 指令就可以了)
Password: (注意喔！這裡打密碼並不會出現 ***** 號)
yenjinc@~$到這裡，就算是登入成功啦！要登出也非常簡單，只要打個 logout 指令就好了！
yenjinc@~$ logout
Connection to yenjinc.info closed. (已經登出囉！)
```

另外，在第一次登入系統之後，最好先更改自己的登入密碼，以免系統預設的密碼太過簡單，而導致系統被入侵。更改密碼的指令是 `passwd`，使用方式在下面介紹。

```
yenjinc@~$ passwd
Changing password for yenjinc
(current) UNIX password: (輸入目前的密碼)
Enter new UNIX password: (輸入新的密碼)
Retype new UNIX password: (再次輸入新的密碼，核對是否輸入正確)
passwd: password updated successfully (密碼更改成功)
```

4.4 認識登入後的環境

在上節裡，我們遠端登入後，會進入到一個終端機的畫面，這一個操作環境稱為「Shell」，使用者 Login 之後，就會來到 Shell 介面，我們可以透過 Shell 下達指令給作業系統，要求系統幫我們做事情，下面一張簡單的圖示來介紹 Shell 概念。各位可以直接在 Shell 下輸入指令，然後 Shell 就會將指令傳達給作業系

統來執行，等到作業系統將工作完成後，就會把執行完成的結果再傳回到 Shell，因此我們就可以立即的看到指令執行完成的結果啦！

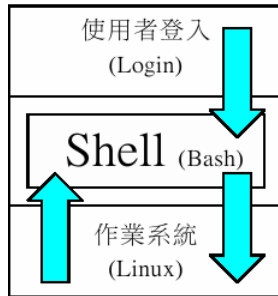


圖 4.2 使用者與 shell 與 Linux 之間的關係

登入之後，各位應該會在畫面上，看到類似這樣的提示畫面「yenjinc@riemann:~\$」，現在我們就來看這些特殊符號代表些什麼意義？

```
yenjinc @ riemann : ~ $
```

簡單的說，「yenjinc@riemann:~\$」，所代表的意義是：[yenjinc (姓名)] [@ (在)] [riemann (主機名稱)] [: (分隔符號)] [~ (所在目錄)] [\$ (提示字元)]，「yenjinc」是使用者的名稱 (username)，「@ (小老鼠)」代表「at (在)」的意思，「riemann」代表所在的「主機名稱」，然後「:」只是一個簡單的分隔符號，「~」這一個位置所代表的意思是「所在目錄」，而上述這個例子「~」所代表的是「家目錄」的意思，最後「\$」符號是「提示字元」的意思，由於這裡「yenjinc」是「一般使用者」不是「管理者 (超級使用者)」於是提示字元為「\$」，超級使用者的提示字元則為「#」符號。

yenjinc	表示「使用者名稱」(username 或 ID)，這個例子是使用者 yenjinc 登入到系統，於是在第一個位置顯示其名稱「yenjinc」。
@	「At (在)」的意思。
riemann	代表所在的「主機名稱」，這個例子表示，使用者 yenjinc 登入到 riemann 的主機上，所以主機名稱欄位的地方，顯示為「riemann」。
:	只是一般的分隔符號，沒有特別的意思。
~	這一個位置所代表的意思是「所在目錄」，而上述這個例子「~」所代表的是「家目錄的意思」，因此你若在其他目錄時，這個地方就會顯示相對應的「其他目錄」的位置。

\$	在 Unix-like / Linux 系統下，有兩種人，一個是「一般使用者」，一個是「超級使用者 (root)」，使用者就屬於一般的人們，而超級使用者就像是「神」！在系統中，超級使用者擁有最大的權限 (就跟天神一般)，可以做任何他想做的事情！在這裡「\$」符號代表的就是一般使用者的提示字元，如果你使用超級使用者登入系統，那個這個提示字元就會轉變成「#」。
----	---

所以我們來看下面這個簡單的舉例：

```

yenjinc @ riemann : ~ $      (yenjinc 一般使用者登入到 riemann 主機，正在家目錄中)
yenjinc @ riemann : /boot $  (yenjinc 一般使用者登入到 riemann 主機，正在 /boot 目錄中)
root @ riemann : ~ #        (root 超級使用者登入到 riemann 主機，正在家目錄中)
root @ riemann : /boot #    (root 超級使用者登入到 riemann 主機，正在 /boot 目錄中)

```

有了這樣的概念後，我們要來看系統預設登入之後，會來到哪裡？預設的情況下，一個使用者登入之後，首先會進入到一個目錄，這個目錄稱作「家目錄」，這是使用者的專屬目錄 (意指：你可以在「你家」做任何你想做的事情)，每個使用者都會有自己的家目錄，彼此不會受到影響，使用 `pwd` 指令可以看到目前所在的目錄位置。

```

yenjinc@~$ pwd
/home/yenjinc          (表示目前正在自己的家目錄當中)

```

4.5 Linux 的基本工具

從後面的章節開始，筆者要來介紹 Command line (命令列模式) 指令的應用囉！在 4.2「遠端登入」的時候提到了，Unix-like / Linux 系統下，最讓人津津樂道的一項特色是「你永遠可以不必在電腦面前使用電腦」，在這裡筆者要再補充另一項特點：「你永遠可以不需要用圖形介面來操作系統」，換句話說，各位可以用「指令 (文字模式)」的方式，來操作系統，完成所有的任務哦！而且使用指令來操作系統，可比使用圖形化介面操作來得更簡潔更有效率呢！

現在就帶領各位來熟悉 Unix / Linux 工具的使用，在第四章會提到的指令分成三個部份：

1. 檔案與目錄管理指令
2. 打包與壓縮指令
3. 更改檔案權限指令

在這裡筆者只會簡單列出指令的「基本使用方式」跟「概念介紹」，並且帶些「實用的範例」來加深各位的使用印象。

4.5.1 檔案與目錄管理

在 Windows 底下，或者任何的 GUI 視窗系統之下，管理檔案跟目錄，不外乎就是使用滑鼠點按來完成這樣的動作。在 Linux 文字模式下，我們要下達「指令」來完成這些事情，使用指令來管理檔案跟目錄是非常有效率的哦！各位可以想像一下，在 GUI 模式下，假使你想將系統上分佈在各個目錄中的檔案（總共有 1000 多個小檔），搬移到一個叫做 tmp 的目錄下，那你大概要花多少時間才能完成這樣的工作？假使是使用指令模式來作這件事的話，只要下達「一行指令」，就可以完成這件事囉！（不相信嗎？呵呵！）

在這部份我們會談到的指令列整如下：

查詢狀態類	
ls	查看目錄內容
pwd	查看目前所在目錄位置
目錄類	
cd	切換目錄
mkdir	建立目錄
rmdir	刪除空目錄（如果目錄裡有資料，要用 rm -R）
檔案類	
touch	建立空檔案（touch 實際的用法是改變檔案建立時間）
rm	刪除檔案（刪除目錄可打 rm -R）
cp	複製檔案（複製目錄可打 cp -R）
mv	搬移檔案或更名
ln	連結檔案（類似建立捷徑的意思）

觀看檔案類	
cat	觀看檔案內容
less	觀看檔案內容（以頁顯示，可以上下捲動）
more	觀看檔案內容（以頁顯示，不能上下捲動）
head	觀看檔案的開頭起始行（預設為前 10 行）
tail	觀看檔案的結尾倒數行（預設為倒數 10 行）
查詢手冊類	
man	查看指令的用法（manpage），也就是人家常說的，有問題就找男人。
info	查看指令的用法（同 manpage，但具有超連結）

在 GUI 圖形介面化的系統中，各位可以直接透過滑鼠的拖拉放以及鍵盤的 Delete 鍵，來完成檔案與目錄的管理，不過在 Unix/Linux 的文字介面當中，各位則必須一些基礎的文字命令，來完成這些任務，首先我們來看「查詢狀態類」與「目錄類」的指令使用方式。

還記得在 4.4 「認識登入後的環境」裡，有約略提到 Shell 提示字元的意義，因此在下面筆者就不再解釋符號的意義囉！這裡我們直接使用應用範例，來介紹指令的使用。

登入到系統時，會看到這樣的提示符號，等待使用者下達指令：

```
cyj@riemann:~$
```

可以使用「man」這個指令，查看指令的說明手冊，這裡查看一下「ls」的使用方法：

```
cyj@riemann:~$ man ls
Reformatting ls(1), please wait...
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
(詳細 manpage 內容，在這裡就留給各位自行 man 囉！)
```

「ls」指令的使用範例：

```

cyj@riemann:~$ ls
File Project bin conf public_html
(使用 ls 會將現在所在目錄的內容顯示出來)

cyj@riemann:~$ pwd
/home/cyj
(使用 pwd 會顯示出目前所在目錄位置)

cyj@riemann:~$ ls /
backup cdrom home lib mnt root sys var
bin dev initrd lost+found opt sbin tmp vmlinuz
boot etc initrd.img media proc srv usr
(查看一下根目錄有些甚麼目錄與檔案)

cyj@riemann:~$ ls -l
total 20
drwxr-xr-x 2 cyj cyj 4096 Aug 7 13:49 File
drwxr-xr-x 9 cyj cyj 4096 Aug 17 15:57 Project
drwxr-xr-x 2 cyj cyj 4096 Aug 13 09:40 bin
drwxr-xr-x 2 cyj cyj 4096 Aug 13 11:33 conf
drwxr-xr-x 2 cyj cyj 4096 Aug 12 09:28 public_html
(加上「-l」的參數，會將顯示狀態變成「完整長格式」，也就是將完整的資訊統統列出來了！)

```

它的格式有「七個欄位」分別是：

權限	連結數	檔案擁有者	檔案擁有群組	檔案大小	最後修改的時間	檔名
----	-----	-------	--------	------	---------	----

每個欄位的代表意義，筆者在後面的章節中，會陸續介紹給各位認識。

```

cyj@riemann:~$ ls -a
. .bash_history .bash_profile .rnd .viminfo Project conf
.. .bash_logout .bashrc .ssh File bin public_html
(加上「-a」的參數後，會把在目錄中的「隱藏檔案」也一併列示出來，那些以「.»為開頭的檔案就是隱藏檔，平常ls假使不加上-a則不會看見這些以.為開頭的隱藏檔。)
```

```

cyj@riemann:~$ ls -al
total 60
drwxr-xr-x 8 cyj cyj 4096 Sep 19 11:11 .

```

```

drwxrwsr-x  39 root  staff 4096 Sep 18 09:26 ..
-rw-----  1 cyj  cyj   142 Sep 21 20:17 .bash_history
-rw-r--r--  1 cyj  cyj    25 Aug  6 16:16 .bash_logout
-rw-r--r--  1 cyj  cyj   602 Aug  4 13:23 .bash_profile
-rw-r--r--  1 cyj  cyj  1834 Aug  4 13:13 .bashrc
-rw-----  1 root  root  1024 Aug 13 09:34 .rnd
drwx-----  2 cyj  cyj  4096 Aug  4 13:14 .ssh
-rw-----  1 cyj  cyj  6683 Sep 19 11:11 .viminfo
drwxr-xr-x  2 cyj  cyj  4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj  cyj  4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj  cyj  4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj  cyj  4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj  cyj  4096 Aug 12 09:28 public_html

```

(OK! 加上 `-a` 與 `-l` 之後，就會完整的以長格式列出目錄中所有的檔案囉！當然，`-a` 與 `-l` 這兩個參數不一定要分開寫，合在一起也行！像是「`-al`」這樣。在 Unix/Linux 當中，所有指令的 Option 參數，都可以像這樣「寫在一起」唷！)

介紹完 `ls` 與 `pwd` 的用法之後，想必各位已經對於文字命令模式的查看目錄狀態，有了一點點的概念了！接著我們要來認識其他有關於「建立/刪除目錄」、「建立/刪除檔案」、「搬移/複製檔案與目錄」的指令囉！

```

cyj@riemann:~$ pwd
/home/cyj
(使用 pwd 查看目前所在目錄)

cyj@riemann:~$ cd /tmp
(使用 cd 指令，進入到 /tmp 目錄)

cyj@riemann:/tmp$ mkdir TEST-DIR
(使用 mkdir 建立一個空目錄，名稱叫做 TEST-DIR)

cyj@riemann:/tmp$ ls -ld TEST-DIR/
drwxr-xr-x  2 cyj  cyj 1024 Oct 15 13:24 TEST-DIR/
(使用 ls -ld ，「-l」表示以完整長格式顯示，「-d」代表只顯示目錄本身的狀態)

cyj@riemann:/tmp$ mkdir TEST-DIR2
(同樣使用 mkdir 建立一個空目錄，名稱叫做 TEST-DIR2)

```

```
cyj@riemann:/tmp$ ls -ld TEST-DIR*
drwxr-xr-x  2 cyj  cyj 1024 Oct 15 13:24 TEST-DIR
drwxr-xr-x  2 cyj  cyj 1024 Oct 15 13:24 TEST-DIR2
(使用萬用字元「*」號，將所有目錄名稱包含 TEST-DIR 的，都以完整長格式狀態顯示)
```

```
cyj@riemann:/tmp$ rmdir TEST-DIR
cyj@riemann:/tmp$ rm -R TEST-DIR2/
(使用 rmdir 可以將一個空目錄刪除，但假使目錄裡有資料時，這時就必須使用加上「-R」來刪除)
```

```
cyj@riemann:/tmp$ ls -ld TEST-DIR*
ls: TEST-DIR*: No such file or directory
(這時候再去 ls 一下，會發現任何有關 TEST-DIR* 的目錄，都已經被我們刪除囉！)
```

```
cyj@riemann:/tmp$ mkdir TEST
(在 /tmp 中，建立 TEST 目錄)
```

```
cyj@riemann:/tmp$ cd TEST/
cyj@riemann:/tmp/TEST$ touch 1 2 3 4 5
(進入到 TEST 目錄，並使用 touch 指令，建立五個檔案，分別是 1 2 3 4 5)
```

```
cyj@riemann:/tmp/TEST$ ls
1 2 3 4 5
(查看一下，的確已經建立五個檔案了！)
```

```
cyj@riemann:/tmp/TEST$ mkdir MP3
cyj@riemann:/tmp/TEST$ ls
1 2 3 4 5 MP3
(在 TEST 目錄中，建立 MP3 目錄，之後 ls 查看是否建立完成)
```

```
cyj@riemann:/tmp/TEST$ mv 1 2 3 MP3/
cyj@riemann:/tmp/TEST$ ls
4 5 MP3
(使用 mv 指令，將 1 2 3 三個檔案搬移到 MP3 目錄中，並檢查是否搬移成功)
```

```
cyj@riemann:/tmp/TEST$ ls MP3/
1 2 3
(查看一下 MP3 目錄，裡面的檔案即是剛才所搬移過去的 1 2 3)
```



```
cyj@riemann:/tmp/TEST$ cp MP3/[123] .
(將 MP3 目錄中，檔名為 1 2 3 的檔案，複製到現行目錄裡)

cyj@riemann:/tmp/TEST$ ls
1 2 3 4 5 MP3
(ls 一下，發現檔案又重新複製回來了！)

cyj@riemann:/tmp/TEST$ cp -R MP3 MP3-backup
cyj@riemann:/tmp/TEST$ ls
1 2 3 4 5 MP3 MP3-backup
(加上 -R 將 MP3 目錄複製為 MP3-backup，複製目錄時需加上 -R 參數，-R 代表遞迴式的複製)

cyj@riemann:/tmp/TEST$ rm -Rf MP3*
cyj@riemann:/tmp/TEST$ ls
1 2 3 4 5
(將所有 MP3* 目錄強制刪除，於是使用『-Rf』，-R 表示遞迴式的刪除，加上 -f 表示強制之意)

cyj@riemann:/tmp/TEST$ cp 5 6
cyj@riemann:/tmp/TEST$ ls
1 2 3 4 5 6
(將 5 複製成 6，之後使用 ls 檢查一下，已經另外 copy 出新的 6 囉！)

cyj@riemann:/tmp/TEST$ rm *
(將現行目錄 /tmp/TEST 裡面所有的資料統統刪除)

cyj@riemann:/tmp/TEST$ cd ..
cyj@riemann:/tmp$ rmdir TEST/
(回到上一層目錄，將 TEST 目錄刪除)
```

以上是基本的檔案與目錄操作範例介紹，接下來，我們要來認識「ln」連結檔案的指令。在這之前，筆者先簡略的介紹一下「inode」的概念，之後各位再來看 ln 這指令時，才會有比較清楚的認識，不然光是 ln 的「硬連結」、「軟連結」，可能就會搞的頭昏腦脹囉！

在 Linux / Unix-like 系統當中，任何的檔案都有一個獨一無二的編號，叫做「Inode」編號！而這個 Inode 編號存放著有關這個檔案的相關資訊，這些相關資訊都可以從「Inode table」找到相對應的內容！各位可以從下面這張圖來了解 Inode 與 Inode table 的關係。

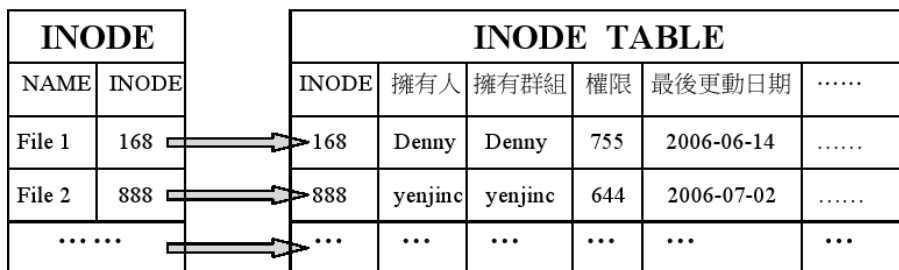


圖 4.3 Inode 與 Inode table 的關係

任何檔案的 inode 都會對應到一個「inode table」，而這個 inode table 就是紀錄著這個檔案的一些相關資訊。從上面的圖例來看，「File 1」的 inode 為「168」，因此將 168 對應到 inode table 時，裡頭就會記載著這個檔案的相關內容，比如檔案的擁有人、擁有群組資訊，相關的檔案權限、或者檔案的最後更動日期...等等。

我們可以使用「ls -li」來查看檔案的 inode 資訊，加上「-li」參數代表顯示 inode 的意思：

```
cyj@riemann:~$ ls -li
total 20
drwxr-xr-x 2 cyj cyj 4096 Aug 13 09:40 bin
drwxr-xr-x 2 cyj cyj 4096 Aug 13 11:33 conf
drwxr-xr-x 2 cyj cyj 4096 Aug 12 09:28 public_html
(使用完整長格式顯示目錄內容)
```

```
cyj@riemann:~$ ls -li
total 20
589829 drwxr-xr-x 2 cyj cyj 4096 Aug 13 09:40 bin
245785 drwxr-xr-x 2 cyj cyj 4096 Aug 13 11:33 conf
2490428 drwxr-xr-x 2 cyj cyj 4096 Aug 12 09:28 public_html
```

檔案的
Inode 資訊

圖 4.4 檔案的 Inode 資訊

(加上「-li」參數，表示要顯示檔案的 Inode 資訊)

有了以上的基本概念後，接著要來看「ln」的使用方式囉！

簡單的說，ln 就像是各位在 Windows 系統中「建立捷徑」的意思，但是又不像 Windows 建立捷徑那樣的單純容易！在 Linux / Unix-like 中，「ln」建立連結有兩種使用方式，一種是建立「硬式連結」，另一種則是建立「軟式連結」。

- 「硬式連結」：檔案並無「來源檔」與「目的檔」的分別，因為都是同樣的 Inode。

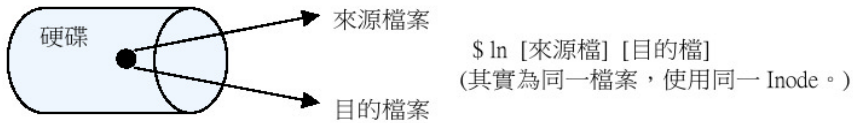


圖 4.5 硬式連結

- 「軟式連結」：檔案有「來源檔」與「目的檔」的分別，兩者屬於不同 Inode。

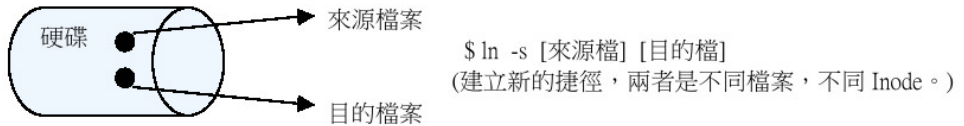



圖 4.6 軟式連結

筆者在下面簡單的示範 ln 的使用方式，並介紹硬式連結與軟式連結的基礎概念。

✎ 「ln」使用方式

```
cyj@riemann:~$ ln -s [來源檔] [目的檔] (加上 -s 參數，表示建立軟連結！)
cyj@riemann:~$ ln [來源檔] [目的檔] (不加上任何參數，表示建立硬連結！)
```

```
cyj@riemann:/tmp/TEST$ touch SourceFile
cyj@riemann:/tmp/TEST$ ls -li
total 0
343394 -rw-r--r-- 1 cyj cyj 0 Oct 15 15:40 SourceFile
(建立 SourceFile 檔案，其 Inode 為 343394)
```


 「建立軟連結」

```

cyj@riemann:/tmp/TEST$ ln -s SourceFile SourceFile-SoftLink
cyj@riemann:/tmp/TEST$ ls -li
total 0
343394 -rw-r--r--  1 cyj cyj  0 Oct 15 15:40 SourceFile
343395 lrwxrwxrwx  1 cyj cyj 10 Oct 15 15:40 SourceFile-SoftLink -> SourceFile

```

(使用 `ln -s` 建立軟連結，建立完成後，產生的「捷徑」檔案會以「->」符號標示著 Source File 為原始檔案，而且兩者的 Inode 並不相同，表示兩個是獨立的個別檔案，並且彼此有相依關係。)


 「建立硬連結」

```

cyj@riemann:/tmp/TEST$ ln SourceFile SourceFile-HardLink
cyj@riemann:/tmp/TEST$ ls -li
total 0
343394 -rw-r--r--  2 cyj cyj  0 Oct 15 15:40 SourceFile
343394 -rw-r--r--  2 cyj cyj  0 Oct 15 15:40 SourceFile-HardLink
343395 lrwxrwxrwx  1 cyj cyj 10 Oct 15 15:40 SourceFile-SoftLink -> SourceFile

```

(建立硬式連結後，可以從上面清楚看到，檔案屬性的「連結數」已經從 1 增加成 2 囉！表示著此檔案目前共有兩個連結數量。另外它們的 Inode 是一樣的，表示其實兩個是一模一樣的檔案，只不過做了一個替身而已唷！)

 「做些簡單的驗證」

```

cyj@riemann:/tmp/TEST$ rm SourceFile
cyj@riemann:/tmp/TEST$ ls -li
total 0
343394 -rw-r--r--  1 cyj cyj  0 Oct 15 15:40 SourceFile-HardLink
343395 lrwxrwxrwx  1 cyj cyj 10 Oct 15 15:40 SourceFile-SoftLink -> SourceFile

```

(將原始的 SourceFile 刪除)

```

cyj@riemann:/tmp/TEST$ cat SourceFile-SoftLink
cat: SourceFile-SoftLink: No such file or directory
cyj@riemann:/tmp/TEST$ cat SourceFile-HardLink

```

(這時候，我們假使查看 SourceFile-SoftLink 的內容時，它會告訴你「找不到檔案」，因為此軟連結的原始檔案已經被刪除囉！當然找不到檔案啦！可是假使去查看 SourceFile-HardLink 的內容時，根本不會發生問題，因為硬連結其實就是同一個檔案，同一個 Inode 建立一個替身罷了！不管你刪除了哪一個替身，對它本身而言並不會有任何的影響唷！)

最後，筆者要來介紹「觀看檔案類」以及「查詢使用手冊類」這些指令的使用範例。在 Linux /Unix-like 系統中，各位可以使用下面這些指令來查看檔案的內容。

觀看檔案類	
cat	觀看檔案內容
less	觀看檔案內容（以頁顯示，可以上下捲動）
more	觀看檔案內容（以頁顯示，不能上下捲動）
head	觀看檔案的開頭起始行（預設為前 10 行）
tail	觀看檔案的結尾倒數行（預設為倒數 10 行）

舉例說明：

```

cyj@riemann:~$ cat /etc/inittab
cyj@riemann:~$ cat /etc/inittab | less
cyj@riemann:~$ less /etc/inittab
cyj@riemann:~$ more /etc/inittab

```

(以上這些方式都可以用來查看檔案的內容，只是差別在於，cat 會一次秀完檔案的內容！而 more 是以頁顯示，但不能上下捲頁！筆者認為最好用的還是 less，各位可以搭配 cat 跟著使用，因為 less 一樣也是以頁顯示，但是可以讓你自由的上下捲頁唷！)

```

cyj@riemann:~$ head /etc/inittab
1. # /etc/inittab: init(8) configuration.
2. # $Id: inittab,v 1.91 2002/01/25 13:35:21 miquels Exp $
3.
4. # The default runlevel.
5. id:2:initdefault:
6.
7. # Boot-time system configuration/initialization script.
8. # This is run first except when booting in emergency (-b) mode.
9. si::sysinit:/etc/init.d/rcS
10.

```

(head 在不加任何參數下，預設為顯示檔案的「前十行」！)

```

cyj@riemann:~$ tail /etc/inittab
1.
2. # Example how to put a getty on a serial line (for a terminal)
3. #
4. #T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
5. #T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100
6.

```

```

7. # Example how to put a getty on a modem line.
8. #
9. #T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3
10.

```

(tail 在不加任何參數下，預設為顯示檔案的「倒數十行」！)

假設目前，我想將 /etc/inittab 的檔案內容，第 13 行到第 18 行的內容秀出來，該如何做呢？

```

cyj@riemann:~$ nl /etc/inittab > TMP

```

(首先將 /etc/inittab 的檔案使用「nl」指令分行後，另儲存為 TMP 檔案)

```

cyj@riemann:~$ head -18 TMP | tail -6

```

```

 13 # Runlevel 0 is halt.
 14 # Runlevel 1 is single-user.
 15 # Runlevels 2-5 are multi-user.
 16 # Runlevel 6 is reboot.
 17 10:0:wait:/etc/init.d/rc 0
 18 11:1:wait:/etc/init.d/rc 1

```

(使用 head 與 tail 複合式的用法，去頭截尾的把 TMP 檔案第 13 - 18 的內容擷取出來！)

以上使用了很多關於「>」轉向、「|」Pipe 等資料流重導向的應用，如果各位還不熟悉這些用法，可以買鳥哥的書來看，或是上鳥哥的網站趕緊補充一些背景知識哦！

由於網路上已有前輩（鳥哥）整理許多不錯的教學文件，在此筆者不打算重新撰寫這些內容，於是採用「提供網址」的方式，讓各位來主動來學習這些資訊與相關內容。

- 「什麼是 Shell？」

http://linux.vbird.org/linux_basic/0320bash.php#bash_what

- 「指令的下達？」

http://linux.vbird.org/linux_basic/0320bash.php#bash_command

- 「絕對路徑與相對路徑」

http://linux.vbird.org/linux_basic/0320bash.php#settings_path

- 「資料流重導向」

http://linux.vbird.org/linux_basic/0320bash.php#redirect

- 「Alias 與 History」

http://linux.vbird.org/linux_basic/0320bash.php#alias_history

- 「認識 Bash Shell 環境」

http://linux.vbird.org/linux_basic/0320bash.php#settings

在 Linux 或其他 Unix-like 系統下，指令的數量是非常的多的！天底下沒有任何人是完完全全能夠記熟所有指令的用法，即使一位系統的老手遇到他不熟的指令時，還是無可奈何。於是在系統底下就有一個「線上求助」，專門在解答這些指令的用法，它就是「manpage」。在 Linux 下，要查看指令的使用說明（manpage）有兩種方式，第一種就是直接 man 指令（例如：man ls）另外一種方式，就是使用 info 指令（例如：info ls）。

假使各位往後碰到不熟悉的指令時，記得叫出「man」與「info」這兩個查詢線上手冊的指令唷！保證「它們」會把所有的使用說明與範例介紹的一清二楚哦！

```
cyj@riemann:~$ man ls      (使用方式為，在 man 之後，加上你想詢問的指令名稱)
cyj@riemann:~$ info ls    (使用方式為，在 info 之後，加上你想詢問的指令名稱)
```

舉個例子來看，假使今天要查看 ls 的用法時：

```
cyj@X40:~> man ls
Reformatting ls(1), please wait...

LS(1)                                                    User Commands LS(1)
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the FILES (the current directory by default). Sort
    entries alphabetically
    if none of -oftuSUX nor --sort.
    Mandatory arguments to long options are mandatory for short options too.
    -a, --all
        do not ignore entries starting with .
    -A, --almost-all
        do not list implied . and ..
```

..... (略)

此時畫面就會立即的顯示出 ls 完整的參數用法還有說明等等。info 的使用方式也是一模一樣！直接「info ls」，就可以看到相關的參數用法與說明囉！從今以後就不用擔心在 Unix-like 系統下不知道怎麼使用指令工具了，因為只要會使用 man 查看 manpage，相信人人都可以成為系統高手哦！

4.5.2 打包與壓縮

打包的概念其實就是「集結檔案」的意思，好比今天各位去 7-11 買了一大堆的飲料跟零食，如果我們不買袋子的話，那就需要很多隻手或者要很多人幫忙拿，可是假使我們用一個大袋子把它們都裝起來後，那就會變得很好拿了！

在檔案處理上，假如你要傳一百個檔案給對方，難道就要一個一個慢慢傳，傳一百次嗎？有沒有什麼方法比較快速又方便呢？是不是可以拿一個大袋子把一百個檔案包起來後，直接傳就好啦！這就是打包的概念。另外如果這一百的檔案包起來後又太大了！那怎麼辦？這時候就需要用到壓縮的工具了！以下就是要來簡單介紹在 Linux / Unix-like 系統下，用來打包與壓縮的工具。

在這部份筆者會提到的指令，列表如下：

打包工具	
tar	將檔案集結（打包起來）
壓縮工具	
gzip	把檔案壓縮
gunzip	把 gzip 檔案解壓縮（同 gzip -d）
bzip2	把檔案壓縮（較佳的壓縮比）
bunzip2	把 bzip2 檔案解壓縮（同 bzip2 -d）
打包檔案與壓縮檔案之副檔名	
tar	*.tar
gzip	*.tar.gz 或 *.tgz
bzip2	*.bz2

我們直接來看操作示範：

「tar 的用法」

```
tar [參數] 檔名 [要打包的檔案]
```

「tar 常用的參數」

-c : create	建立包裝檔 (打包)
-x : extract	解開包裝檔
-v : verbose	顯示詳細資訊
-f : file	指定包裝檔名
-t : list	顯示檔案的內容
-C : change directory	切換到目錄中
-z : gzip	支援 gzip 壓縮或解壓縮
-j : bzip2	支援 bzip2 壓縮或解壓縮

「打包檔案」

```
cyj@riemann:~$ ls -l
total 20
drwxr-xr-x  2 cyj cyj 4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj 4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj 4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj 4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj 4096 Aug 12 09:28 public_html
cyj@riemann:~$ tar -cvf tmp-file.tar bin conf public_html
(tar [參數] 檔名 [要打包的檔案])
bin/
bin/tmp.sh
conf/
conf/httpd.conf
conf/httpd-ssl.conf
conf/vsftpd.conf
conf/vsftpd.conf.ori
conf/vsftpd.pem
conf/vsftpd.user_list
public_html/
public_html/index.php
cyj@riemann:~$ ls -l
```

```
total 124
drwxr-xr-x  2 cyj cyj  4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj  4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj  4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj  4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj  4096 Aug 12 09:28 public_html
-rw-r--r--  1 cyj cyj 102400 Oct 15 22:56 tmp-file.tar
```

(常用的打包參數為「-cvf」，上述例子即是將 bin conf public_html 打包為 tmp-file.tar 檔案)

```
cyj@riemann:~$ ls -l
total 124
drwxr-xr-x  2 cyj cyj  4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj  4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj  4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj  4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj  4096 Aug 12 09:28 public_html
-rw-r--r--  1 cyj cyj 102400 Oct 15 22:56 tmp-file.tar
cyj@riemann:~$ mkdir TMP
(建立新的 TMP 目錄)

cyj@riemann:~$ ls -al TMP/
total 8
drwxr-xr-x  2 cyj cyj 4096 Oct 15 23:03 .
drwxr-xr-x  9 cyj cyj 4096 Oct 15 23:03 ..
(目前 TMP 目錄是空的)
```

「解包檔案」

```
cyj@riemann:~$ tar -xvf      tmp-file.tar -C      TMP
                (tar [參數] 檔名      [參數] 目錄)
                (解包之前，首先進入到 TMP 目錄中，也就是把檔案解包至 TMP 目錄裡的意思)
```

```
bin/
bin/tmp.sh
conf/
conf/httpd.conf
conf/httpd-ssl.conf
conf/vsftpd.conf
conf/vsftpd.conf.ori
conf/vsftpd.pem
```

```

conf/vsftpd.user_list
public_html/
public_html/index.php
cyj@riemann:~$ ls TMP/
bin  conf  public_html
(常用的解包參數為「-xvf」，另外此例加上「-C」參數，表示解包之前首先進入到 TMP 目錄中)

```

「顯示包裝檔案內容」

```

cyj@riemann:~$ ls
File Project bin  conf  public_html  tmp-file.tar
cyj@riemann:~$ tar -tf      tmp-file.tar
                    (tar [參數] 檔名)

bin/
bin/tmp.sh
conf/
conf/httpd.conf
conf/httpd-ssl.conf
conf/vsftpd.conf
conf/vsftpd.conf.ori
conf/vsftpd.pem
conf/vsftpd.user_list
public_html/
public_html/index.php
(使用「-tf」參數，可以查看包裝裡面的檔案資訊)

cyj@riemann:~$ ls
File Project bin  conf  public_html  tmp-file.tar
(僅是顯示檔案內容，並未真正解開包裝，所以ls之後並不會找到解包的檔案)

```

以上是基礎的 tar 打包指令的使用範例，接下來我們來看壓縮工具的使用範例囉！

「gzip 壓縮」

```

cyj@riemann:~$ ls -l
total 124
drwxr-xr-x  2 cyj cyj   4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj   4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj   4096 Aug 13 09:40 bin

```

```
drwxr-xr-x  2 cyj cyj   4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj   4096 Aug 12 09:28 public_html
-rw-r--r--  1 cyj cyj 102400 Oct 15 22:56 tmp-file.tar
```

```
cyj@riemann:~$ gzip tmp-file.tar
( gzip 之後之直接加上要壓縮的檔名就可以囉! )
```

```
cyj@riemann:~$ ls -l
total 52
drwxr-xr-x  2 cyj cyj   4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj   4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj   4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj   4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj   4096 Aug 12 09:28 public_html
-rw-r--r--  1 cyj cyj 32055 Oct 15 22:56 tmp-file.tar.gz
(可以看到，壓縮後的檔案副檔名變成了 .gz，另外檔案的大小也變小許多了)
```

「gzip 解壓縮」

```
cyj@riemann:~$ ls -l
total 52
drwxr-xr-x  2 cyj cyj   4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj   4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj   4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj   4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj   4096 Aug 12 09:28 public_html
-rw-r--r--  1 cyj cyj 32055 Oct 15 22:56 tmp-file.tar.gz
```

```
cyj@riemann:~$ gzip -d tmp-file.tar.gz
(加上「-d」參數，就可以解壓縮囉！其實也可以使用「gunzip tmp-file.tar.gz」，也是一樣的效果哦！)
```

```
cyj@riemann:~$ ls -l
total 124
drwxr-xr-x  2 cyj cyj   4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj   4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj   4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj   4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj   4096 Aug 12 09:28 public_html
```

```
-rw-r--r-- 1 cyj cyj 102400 Oct 15 22:56 tmp-file.tar  
(原本的壓縮檔案已經復原了!)
```

「bzip2 壓縮」

```
cyj@riemann:~$ ls -l  
total 124  
drwxr-xr-x 2 cyj cyj 4096 Aug 7 13:49 File  
drwxr-xr-x 9 cyj cyj 4096 Aug 17 15:57 Project  
drwxr-xr-x 2 cyj cyj 4096 Aug 13 09:40 bin  
drwxr-xr-x 2 cyj cyj 4096 Aug 13 11:33 conf  
drwxr-xr-x 2 cyj cyj 4096 Aug 12 09:28 public_html  
-rw-r--r-- 1 cyj cyj 102400 Oct 15 22:56 tmp-file.tar  
  
cyj@riemann:~$ bzip2 tmp-file.tar  
( bzip2 之後之直接加上要壓縮的檔名就可以囉! )  
  
cyj@riemann:~$ ls -l  
total 44  
drwxr-xr-x 2 cyj cyj 4096 Aug 7 13:49 File  
drwxr-xr-x 9 cyj cyj 4096 Aug 17 15:57 Project  
drwxr-xr-x 2 cyj cyj 4096 Aug 13 09:40 bin  
drwxr-xr-x 2 cyj cyj 4096 Aug 13 11:33 conf  
drwxr-xr-x 2 cyj cyj 4096 Aug 12 09:28 public_html  
-rw-r--r-- 1 cyj cyj 22664 Oct 15 22:56 tmp-file.tar.bz2  
(可以看到，壓縮後的檔案副檔名變成了 .bz2，值得注意的是，採用 bzip2 壓縮檔案的大小會比 gzip  
壓縮檔案要來得更小一些哦!)
```

「bzip2 解壓縮」

```
cyj@riemann:~$ ls -l  
total 52  
drwxr-xr-x 2 cyj cyj 4096 Aug 7 13:49 File  
drwxr-xr-x 9 cyj cyj 4096 Aug 17 15:57 Project  
drwxr-xr-x 2 cyj cyj 4096 Aug 13 09:40 bin  
drwxr-xr-x 2 cyj cyj 4096 Aug 13 11:33 conf  
drwxr-xr-x 2 cyj cyj 4096 Aug 12 09:28 public_html  
-rw-r--r-- 1 cyj cyj 22664 Oct 15 22:56 tmp-file.tar.bz2  
  
cyj@riemann:~$ bzip2 -d tmp-file.tar.bz2
```

(加上「-d」參數，就可以解壓縮囉！其實也可以使用「bunzip2 tmp-file.tar.bz2」，也是一樣的效果哦！)

```
cyj@riemann:~$ ls -l
total 124
drwxr-xr-x  2 cyj cyj  4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj  4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj  4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj  4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj  4096 Aug 12 09:28 public_html
-rw-r--r--  1 cyj cyj 102400 Oct 15 22:56 tmp-file.tar
```

(原本的壓縮檔案已經復原了！)

tar 與 gzip / bzip2 的搭配用法：

```
cyj@riemann:~$ ls
File Project bin conf public_html tmp-file.tar
```

```
cyj@riemann:~$ tar -czvf gzip-tar-file.tgz public_html
public_html/
public_html/index.php
```

(在 tar 的參數中，加上「-z」表示讓打包時直接以 gzip 壓縮)

```
cyj@riemann:~$ tar -cjvf bzip2-tar-file.bz2 public_html
public_html/
public_html/index.php
```

(在 tar 的參數中，加上「-j」表示讓打包時直接以 bzip2 壓縮)

```
cyj@riemann:~$ ls
File      bin                conf                public_html
Project  bzip2-tar-file.bz2  gzip-tar-file.tgz  tmp-file.tar
```

```
cyj@riemann:~$ mkdir TMP1 TMP2
```

```
cyj@riemann:~$ tar -zxvf gzip-tar-file.tgz -C TMP1
```

(將檔案解壓縮至 TMP1 目錄中)

```
public_html/
public_html/index.php
```

(同樣的，在解包的過程中，可以加上 -z 參數，使 tar 可以直接針對 .gz / .tgz 的包裝解壓縮！)

```

cyj@riemann:~$ tar -jxvf bzip2-tar-file.bz2 -C TMP2
          (將檔案解壓縮至 TMP2 目錄中)
public_html/
public_html/index.php
(同樣的，在解包的過程中，可以加上 -j 參數，使 tar 可以直接針對 .bz2 的包裝解壓縮！)

cyj@riemann:~$ ls TMP*
TMP1:
public_html

TMP2:
public_html
(驗證一下，剛才上面的兩個解壓縮的例子，順利成功！ :-D)

```

其實還有更快的入門方式，那就是直接看 manpage 啦！然後在實地的練習一番，想必就能熟通這些工具的用法了 :-P

```

cyj@riemann:~$ man tar
cyj@riemann:~$ man gzip
cyj@riemann:~$ man bzip2

```

4.5.3 更改檔案權限

在 Unix-like 系統下，每一個檔案都有其相關的屬性，而這些屬性背後所代表的意義也是一門重要的課題。

在這部份筆者會提到的指令，列表如下：

更改檔案權限	
chmod	更改檔案的存取權限
chown	更改檔案的擁有者或擁有群組
chgrp	更改檔案的擁有群組

還記得在「4.5.1 章節」已經有約略的介紹「ls -l」在完整長格式的顯示中，各個欄位的意思，現在我們要來了解一下第一個欄位：「權限」的概念。

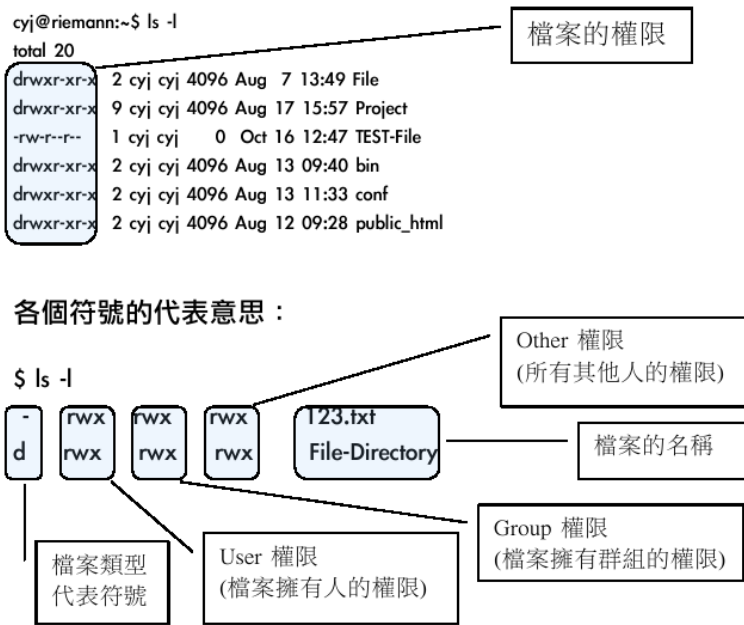


圖 4.7 權限概念示意圖

檔案類型代表符號	
-	Regular Files (表示一般的檔案)
d	Directory (目錄)
l	Link Files (連結檔案)

在 Linux / Unix-like 系統下，任何檔案都有其相關的「權限屬性」，在此權限屬性又細分為三種：

1. **User 權限**：規定著擁有檔案的「擁有人」對於該檔案能做哪些事情。
2. **Group 權限**：規定著擁有檔案的「群組使用者」對於該檔案能做哪些事情。
3. **Other 權限**：規定「其他所有人」對於該檔案能做哪些事情。

每一個權限又各別由「r」、「w」、「x」三個符號所組成，筆者將符號的代表意義整理如下表：

使用權限	符號	對於檔案	對於目錄
讀取	r	檢視檔案內容	列示目錄的內容
寫入	w	寫入 / 變更檔案	建立 / 移除 目錄中的檔案
執行	x	將檔案當成程式來執行	允許切換 (cd) 到目錄中

對於檔案而言：

一個檔案假使具有「r」的權限，表示各位可以用任何的工具來「讀取」檔案的內容。如果它擁有「w」的權限，那麼各位可以對於該檔案的內容進行「任何的修改」。若該檔案具有「x」的權限，那麼這個檔案可以看作為「可執行」檔案，可視為程式來執行。

對於目錄而言：

一個目錄假使具有「r」的權限，表示各位可以在此目錄中使用像是 ls 等指令來「列示」出目錄的內容。如果它擁有「w」的權限，那麼各位可以對於該目錄裡面的檔案進行「任何的處理」，表示你可以「建立」或「刪除」目錄中的檔案。若一個目錄具有「x」的權限，這表示各位被允許切換 (進入) 到該目錄中，換句話說，一個目錄若不賦予「x」的權限的話，那麼各位將無法進入 (cd) 到目錄之中唷！

我們來看一些示範的例子：

```
cyj@riemann:~$ ls -l
total 20
drwxr-xr-x  9 cyj cyj 4096 Aug 17 15:57 Project
-rw-r--r--  1 cyj cyj    0 Oct 16 12:47 TEST-File
```

(這裡可以看到一個 Project 目錄，一個 TEST-File 檔案。)

Project 目錄的屬性

```
User :   rwx   可讀，可寫入，可執行。
Group :  r-x   可讀，不可寫入，可執行。
Other :  r-x   可讀，不可寫入，可執行。
```

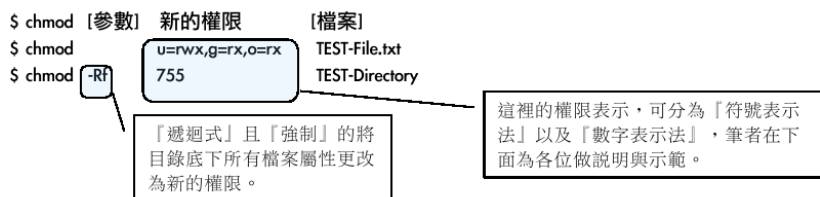
(這表示 Project 目錄，「cyj 使用者」擁有最大的權限，在此目錄中 cyj User 「可以做任何事情」！而「cyj 群組 (Group)」在此目錄中，只可以「讀取」以及「進入」，並不賦予任何有關「寫入」的權限。最後「其他任何人」在此目錄中，同樣也只有「讀取」跟「進入」的權限而已，也無法做任何「寫入」的動作。)

TEST-File 檔案的屬性

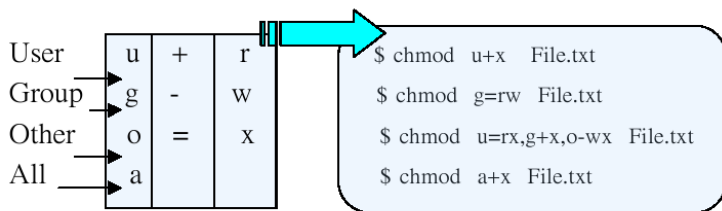
User : rw- 可讀，可寫入，不可執行。
 Group : r-- 可讀，不可寫入，不可執行。
 Other : r-- 可讀，不可寫入，不可執行。

(這表示 TEST-File 檔案，「cyj 使用者 (User)」可以針對此檔案有「讀取」與「寫入/修改」的權限，但是並不賦予「可執行」的權限！而「cyj 群組 (Group)」針對此檔案，僅僅可以「讀取」，並不賦予任何有關「寫入」以及「執行」的權限。最後「其他任何人」對於此檔案，同樣也只有「讀取」的權限而已，也無法做任何「寫入」與「執行」的動作。)

OK！初步來講，各位有這樣基本的檔案權限概念就可以囉！接著筆者要來介紹在 Linux / Unix-Like 系統中，用來「更改檔案權限」的工具：「chmod」的使用概念。



符號表示法



數字表示法

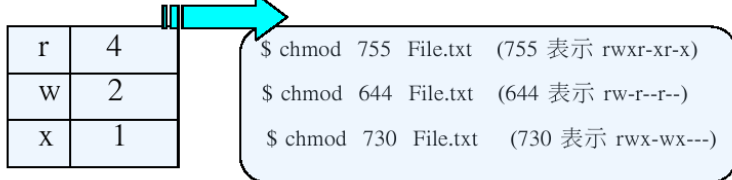


圖 4.8 chmod 的基礎用法

我們來看一些示範的例子：

```
cyj@riemann:~/TMP$ ls -l
total 0
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 A
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 B
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 C
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 D
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 E
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 F
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 G
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 H
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 I
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:12 J
(原始的檔案權限資訊)

cyj@riemann:~/TMP$ chmod 742 A
cyj@riemann:~/TMP$ chmod 135 B
cyj@riemann:~/TMP$ chmod 246 C
cyj@riemann:~/TMP$ chmod 341 D
cyj@riemann:~/TMP$ chmod 271 E
cyj@riemann:~/TMP$ chmod u=rw,g=x,o=wx F
cyj@riemann:~/TMP$ chmod g-wx,o+x G
cyj@riemann:~/TMP$ chmod a-wx H
cyj@riemann:~/TMP$ chmod u+rx,o-x I
cyj@riemann:~/TMP$ chmod a-rwx J
(分別採用『數字表示法』以及『符號表示法』來變更權限)

cyj@riemann:~/TMP$ ls -l
total 0
-rwxr---w- 1 cyj cyj 0 Oct 16 20:12 A
---x-wxr-x 1 cyj cyj 0 Oct 16 20:12 B
--w-r--rw- 1 cyj cyj 0 Oct 16 20:12 C
--wxr----x 1 cyj cyj 0 Oct 16 20:12 D
--w-rwx--x 1 cyj cyj 0 Oct 16 20:12 E
-rw---x-wx 1 cyj cyj 0 Oct 16 20:12 F
-rw-r--r-x 1 cyj cyj 0 Oct 16 20:12 G
-r--r--r-- 1 cyj cyj 0 Oct 16 20:12 H
```

```
-rwxr--r-- 1 cyj cyj 0 Oct 16 20:12 I
----- 1 cyj cyj 0 Oct 16 20:12 J
(變更完成後的狀態)
```

了解 chmod 以及權限的概念後，接著，我們要來認識以下這兩個指令的使用方式：

chown	更改檔案的擁有者或擁有群組
chgrp	更改檔案的擁有群組

藉由剛才 chmod 的使用介紹，大家對於第一個欄位「權限」已經有基礎的概念囉！現在我們要來看第三與第四個欄位：「檔案擁有人」、「檔案擁有群組」。

```
cyj@riemann:~$ ls -l
total 20
drwxr-xr-x 2 cyj cyj 4096 Aug 7 13:49 File
drwxr-xr-x 9 cyj cyj 4096 Aug 17 15:57 Project
-rw-r--r-- 1 cyj cyj 0 Oct 16 12:47 TEST-File
drwxr-xr-x 2 cyj cyj 4096 Aug 13 09:40 bin
drwxr-xr-x 2 cyj cyj 4096 Aug 13 11:33 conf
```

第一行：
檔案擁有人

第二行：
檔案擁有群組

圖 4.9

在系統當中，每一個檔案都會有擁有者與擁有群組，我們可以從家目錄 (/home) 中清楚的看出來。在系統家目錄當中，每一位使用者都是他們自己家目錄的擁有者以及擁有群組。

```
cyj@riemann:~$ ls -l /home/
total 220
drwxr-xr-x 5 apua apua 4096 Oct 16 11:19 apua
drwxr-xr-x 3 bill bill 4096 Sep 18 09:55 bill
drwxr-xr-x 4 bluelight bluelight 4096 Sep 18 10:11 bluelight
drwxr-xr-x 4 board board 4096 Aug 24 12:01 board
drwxr-xr-x 3 chang chang 4096 Aug 11 13:16 chang
drwxr-xr-x 3 chenmvm0 chenmvm0 4096 Aug 11 13:16 chenmvm0
drwxr-xr-x 9 cyj cyj 4096 Oct 16 20:12 cyj
drwxr-xr-x 7 czhsu czhsu 4096 Oct 9 22:57 czhsu
```

擁有者

擁有群組

圖 4.10

而「chown」以及「chgrp」這兩個指令即是用來變更檔案目錄之「擁有者」或「群組」的工具。此兩種 Unix 工具的使用方式，筆者由下方的使用範例來為各位做說明。

chown 用法

```
chown [擁有者] : [擁有群組] 檔案
```

chgrp 用法

```
chgrp [擁有群組] 檔案
```

應用範例介紹

```
cyj@riemann:/tmp/TEST$ ls -l
total 0
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:46 File1
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:46 File2
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:46 File3
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:47 File4
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:47 File5
(目前這五個檔案 File1~File5 的擁有者與群組都屬於 cyj )

cyj@riemann:/tmp/TEST$ chown cyj:yenlung File1
chown: changing ownership of `File1': Operation not permitted
(由於「更改權限」以及「更改群組」已經不屬於一般使用者能做的事情囉！所以這邊會有「權限不足」的警告訊息！)

cyj@riemann:/tmp/TEST$ su -
Password: (輸入 root 密碼)
(可以使用「su -」這個指令來更改自己的權限，晉升為「超級使用者 root」)

riemann:~# cd /tmp/TEST/
(目前登入的身份已經轉變為「超級使用者」囉！因此提示字元也變成「#」符號囉！)

riemann:/tmp/TEST# ls -l
total 0
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:46 File1
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:46 File2
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:46 File3
-rw-r--r-- 1 cyj cyj 0 Oct 16 20:47 File4
```

```
-rw-r--r--  1 cyj cyj 0 Oct 16 20:47 File5

riemann:/tmp/TEST# chown cyj:yenlung File1
(將 File1 的擁有者更改為 cyj，擁有群組更改為 yenlung )

riemann:/tmp/TEST# chown yenlung File2
(將 File2 的擁有者更改為 yenlung )

riemann:/tmp/TEST# chown :kkd File3
(將 File3 的擁有群組更改為 kkd )

riemann:/tmp/TEST# chgrp apua File4
(使用「chgrp」將 File4 的擁有群組更改為 apua，其實它的用法等同於：「chown :apua File4」)

riemann:/tmp/TEST# chown yenlung:apua File5
(將 File5 的擁有者更改為 yenlung，擁有群組更改為 apua )

riemann:/tmp/TEST# ls -l
total 0
-rw-r--r--  1 cyj          yenlung   0 Oct 16 20:46 File1
-rw-r--r--  1 yenlung    cyj       0 Oct 16 20:46 File2
-rw-r--r--  1 cyj        kkd        0 Oct 16 20:46 File3
-rw-r--r--  1 cyj        apua        0 Oct 16 20:47 File4
-rw-r--r--  1 yenlung    apua      0 Oct 16 20:47 File5
(驗證一下，剛才的更改動作全部成功啦！)

riemann:/tmp/TEST# chgrp jimyeh File*
(最後筆者使用 chgrp 將所有檔案的擁有群組統統改成 jimyeh )

riemann:/tmp/TEST# ls -l
total 0
-rw-r--r--  1 cyj          jimyeh   0 Oct 16 20:46 File1
-rw-r--r--  1 yenlung    jimyeh   0 Oct 16 20:46 File2
-rw-r--r--  1 cyj          jimyeh   0 Oct 16 20:46 File3
-rw-r--r--  1 cyj          jimyeh   0 Oct 16 20:47 File4
-rw-r--r--  1 yenlung    jimyeh   0 Oct 16 20:47 File5
(驗證一下，剛才的更改動作全部成功啦！)
```

4.5.4 尋找檔案與內容

尋找系統上的檔案

在 Linux 系統當中，檔案的數量非常非常的多，假使各位今天要找一個叫做「RELEASE-NOTES.zh_TW.html」的網頁，如果沒有使用特殊的搜尋工具，那豈不是要翻遍系統才能找到它。幸好，在系統已經內建許多搜尋檔案的工具給我們使用啦！下面我們要來簡單介紹這些工具的使用方法。

在這裡會使用到的指令，筆者把它們整理列表如下：

搜尋系統檔案	
find	搜尋系統中的檔案
locate	搜尋系統中的檔案（透過已整理的資料庫）
whereis	搜尋指令名稱的相關檔案
which	搜尋指令名稱所存在的位置（根據 PATH）
type	查詢指令名稱所存在的位置以及種類

「find」與「locate」都是可以直接用來搜尋檔案的工具，不過比較特別的是，find 的搜尋方式，是直接從 File System 來搜尋，因此搜尋的速度會比較慢，而且也比較耗費系統資源。locate 的搜尋方式，是透過一個已經索引（index）完成的資料庫來找出檔案所在位置，所以 locate 搜尋的速度比較快速，也不會耗費太多的系統資源，不過通常在使用 locate 之前，都要先使用「updatedb」指令，來建立檔案索引資料庫。

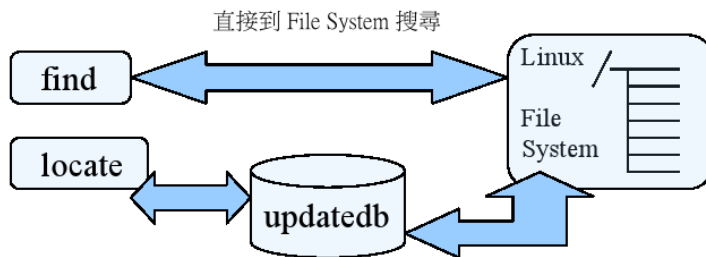


圖 4.11

除了搜尋系統的檔案之外，在此也要介紹各位有關於「指令名稱」或「指令相關資料」的搜尋工具，例如 `whereis`、`which`、`type` 等等。下面筆者直接以一些常用的使用範例來為各位介紹這些 Unix 工具的使用方法。

find 用法

```
$ find [目錄] [參數] 要搜尋的字串 [動作]
$ find /home -name "yenjinc.info"
$ find /home -name "yenjinc.info" -exec rm {} \;
```

使用 find 搜尋檔案

```
cyj@riemann:~$ ls -l
total 24
drwxr-xr-x  2 cyj cyj 4096 Aug  7 13:49 File
drwxr-xr-x  9 cyj cyj 4096 Aug 17 15:57 Project
drwxr-xr-x  2 cyj cyj 4096 Oct 16 20:12 TMP
drwxr-xr-x  2 cyj cyj 4096 Aug 13 09:40 bin
drwxr-xr-x  2 cyj cyj 4096 Aug 13 11:33 conf
drwxr-xr-x  2 cyj cyj 4096 Aug 12 09:28 public_html
cyj@riemann:~$ find . -name "Test.txt"
./Project/ssh/Test.txt
(一個典型的搜尋檔案的範例)
```

使用 find 搜尋檔案並執行相關動作

```
cyj@riemann:~$ find . -name "Test.txt"
./Project/ssh/Test.txt
(找到 Test.txt 檔案在 ./Project/ssh/ 中)

cyj@riemann:~$ find . -name "Test.txt" -exec rm {} \;
```

(加上「`-exec`」表示在搜尋之後即將「執行相關動作」，這一個例子是：當找到檔案後，對這些檔案做刪除的動作，於是在 `-exec` 之後緊接著是「`rm {} \;`」，「`rm`」是刪除檔案的意思，而「`{}`」則表示搜尋到的檔案 (`Test.txt`)，最後的「`\;`」表示此動作的結束。)

```
cyj@riemann:~$ find . -name "Test.txt"
(由於該檔已經被刪除了，所以現在已經搜尋不到囉！)
```


「-atime」、「-ctime」、「-mtime」的應用

```

cyj@riemann:~$ man find
Reformatting find(1), please wait...

-atime n
        File was last accessed n*24 hours ago.
        (Access: 表示「存取」、「Read」)

-ctime n
        File's status was last changed n*24 hours ago.
        (Change: 改變過狀態, 例如: 「改名」、「修改權限」..等)

-mtime n
        File's data was last modified n*24 hours ago.
        (Modify: 表示「修改過內容」)

cyj@riemann:/$ cd /
cyj@riemann:/$ date
Tue Oct 17 21:38:47 CST 2006
(使用「date」指令查看當下的日期與時間)

cyj@riemann:/$ su -
Password:
(由於接下來針對部份系統目錄的搜尋, 需要特殊權限, 所以先切換成超級使用者 root)

riemann:~# find /var -mtime 10 -exec ls -l {} \;
-rw----- 1 root root 2477 Oct  7 07:55 /var/log/vsftpd.log.2
(針對「/var」目錄來找出「10天之前」任何有「修改過」的檔案, 並且使用 ls -l 將其顯示出來。各位可以從 ls -l 完整長格式中發現, 該檔案在「Oct 7」時最後被更動過 (確實是符合10天之前的條件)。)

riemann:~# find /var -atime 5 -exec ls -l {} \;
-rw-r----- 1 Debian-exim adm 677 Oct 12 06:25 /var/log/exim4/mainlog.7.gz
-rw-r----- 1 root adm 851 Oct 13 06:25 /var/log/syslog.4.gz
-rw-rw---- 1 mysql adm 887461 Oct 13 06:25 /var/log/mysql/mysql-bin.073
-rw-r----- 1 root adm 20 Oct 12 06:25 /var/log/mysql.err.6.gz
-rw-r----- 1 mysql adm 20 Oct 12 06:25 /var/log/mysql.log.6.gz
-rw-rw---- 1 apua mail 0 Oct 12 12:22 /var/mail/apua
(針對「/var」目錄來找出「5天之前」任何有「存取過」的檔案, 並且使用 ls -l 將其顯示出來。各位可以從 ls -l 完整長格式中發現, 這些檔案的最後更動日期的確都是在「Oct 17」起始的這五天之內。)
```

```
riemann:~# find /var -ctime 1 -exec ls -l {} \;
-r--r--r-- 1 root root 79856 Oct 16 06:25 /var/log/ksymoops/20061016062552.ksyms
-r--r--r-- 1 root root 985 Oct 16 06:25 /var/log/ksymoops/20061016062552.modules
-rw-r----- 1 root adm 6689 Oct 16 08:32 /var/log/apache-ssl/error.log
-rw-rw---- 1 mysql adm 450785 Oct 16 06:25 /var/log/mysql/mysql-bin.076
-rw----- 1 root root 689 Oct 15 22:10 /var/log/vsftpd.log
```

(針對「/var」目錄來找出「1 天之前」任何有「改變過狀態」的檔案，並且使用 `ls -l` 將其顯示出來。各位可以從 `ls -l` 完整長格式中發現，這些檔案的最後更動日期的確都是在「Oct 17」起始的這一天之內。)

「-amin」、「-cmin」、「-mmin」的應用

```
cyj@riemann:~$ man find
Reformatting find(1), please wait...

-amin n
        File was last accessed n minutes ago.
        (Access: 表示「存取」、「Read」)

-cmin n
        File's status was last changed n minutes ago.
        (Change: 改變過狀態，例如：「改名」、「修改權限」..等)

-mmin n
        File's data was last modified n minutes ago.
        (Modify: 表示「修改過內容」)
```

「-amin」、「-cmin」、「-mmin」這些參數的使用方式與上述的「-atime」、「-ctime」、「-mtime」的使用，基本上用法是大同小異唷！假使各位有興趣的話，不妨切換到 /tmp 目錄底下，實際的利用 `find` 加上這些參數來試試看！以上這些參數在現實生活中很常用到唷！假使各位親身體驗過這些參數的使用後，一定會更加深印象的！筆者猜想，總有一天，你一定會用到它的！ :-D

locate 用法

```
$ locate [找尋的檔案名稱]
$ locate TEST.File.txt
```

在使用 locate 之前，必須先更新 locate 資料庫，使用「updatedb」指令

```

cyj@riemann:~$ sudo updatedb
(「sudo」指令可以讓此次執行的動作是以 root 權限來執行，在此用 locate 之前，記得要先執行「updatedb」，此命令會先將 File System 的檔案資料索引，並存放在「/var/cache/locate/locatedb」)

cyj@riemann:~$ locate passwd
/etc/passwd
/etc/passwd-
(找尋名稱為 passwd 的檔案)

cyj@riemann:~$ locate httpd.conf
/home/cyj/LocalApps/Apache/conf/httpd.conf
/home/cyj/LocalApps/Apache/conf/original/httpd.conf
(找尋名稱為 httpd.conf 的檔案)

```

locate 的使用方式要比 find 來得簡單，而且在搜尋檔案的速度上，也較 find 要來得快速，不過 locate 有個小小缺點就是，各位的系統要「定期」的「updatedb」，這樣才能讓檔案索引的內容能一直保持在最新的狀態，假使各位在 updatedb 之後，才建立新的檔案，那麼這個新的檔案有可能會搜尋不到唷！這是因為 updatedb 資料庫尚未更新的原因。

以下是筆者針對此狀況所示範的簡單例子：

```

cyj@riemann:~$ ls
Desktop  Documents Library  Movies  Music  Pictures  Public  Sites

cyj@riemann:~$ touch New-File.txt
cyj@riemann:~$ ls
Desktop      Library      Music      Pictures      Sites
Documents    Movies      New-File.txt  Public
(建立一個新的檔案)

cyj@riemann:~$ locate New-File.txt
(這時候，筆者直接使用 locate 來搜尋這個新建立的檔案，居然找不到任何資料！)

cyj@riemann:~$ sudo updatedb

```

(於是筆者手動重新在 updatedb 一次)

```
cyj@riemann:~$ locate New-File.txt
/home/cyj/New-File.txt
```

(這時候，剛才新建立的檔案已經被索引到資料庫中了，所以可以順利找到囉！所以各位的系統要「定期」的「updatedb」，這樣才能讓檔案索引的內容能一直保持在最新的狀態喲！)

看完了 find、locate 的用法之後，接下來要介紹各位有關於「指令名稱」或「指令相關資料」的搜尋工具，例如 whereis、which、type 等等。下面筆者直接以一些常用的使用範例來為各位介紹這些 Unix 工具的使用方法。

whereis	搜尋指令名稱的相關檔案
which	搜尋指令名稱所存在的位置 (根據 PATH)
type	查詢指令名稱所存在的位置以及種類

「whereis 用法」

```
$ whereis [參數] 查詢的檔案名稱
$ whereis -b man
$ whereis man
```

查詢指令名稱的相關檔案

```
-b Search only for binaries.
-m Search only for manual sections.
-s Search only for sources.


cyj@riemann:~$ whereis -b man
man: /usr/bin/man /usr/local/man /usr/share/man
(找尋有關 man 的 binary 檔案)

cyj@riemann:~$ whereis -m man
man: /usr/share/man/man1/man1.gz /usr/share/man/man7/man7.gz
(找尋有關 man 的 manual 使用手冊檔案)

cyj@riemann:~$ whereis -s man
(找尋有關 man 的 source 檔案，由於系統中沒有 man 的 source File 所以在此搜尋不到)
```

 「which 用法」

```
$ which [查詢的指令名稱]
$ which ls
$ which mkdir
```

 查詢指令檔案的所在位置 (根據 PATH)


```
cyj@riemann:~$ which ls
/bin/ls
(根據 PATH 所找到的 ls 所在位置在 /bin/ls)

cyj@riemann:~$ which man
/usr/bin/man
(根據 PATH 所找到的 man 所在位置在 /usr/bin/man)

cyj@riemann:~$ which mkdir
/bin/mkdir
(根據 PATH 所找到的 mkdir 所在位置在 /bin/mkdir)
```

 「type 用法」

```
$ type [查詢的指令名稱]
$ type mkdir
```

 查詢指令名稱所存在的位置以及種類

```
cyj@riemann:~$ type man
man is hashed (/usr/bin/man)
(查詢的結果表示, man 是在 /usr/bin/man)

cyj@riemann:~$ type mkdir
mkdir is /bin/mkdir
(查詢的結果表示, mkdir 是在 /bin/mkdir)

cyj@riemann:~$ type cd
cd is a shell builtin
(查詢的結果表示, cd 是 shell 內建的指令)

cyj@riemann:~$ type ls
ls is aliased to `ls --color=auto'
(查詢的結果表示, ls 在此是一個「Alias」, 其內容是: 「ls --color=auto」)
```

尋找檔案的內容

筆者在上一章節裡所介紹的都是搜尋檔案的「名稱」，現在要來介紹的是搜尋檔案的「內容」（也就是「擷取檔案字串內容」的意思）。換句話說，比如今天大家要搜尋系統中所有的 *.txt 檔案，要來找出哪幾個 txt 檔案的內文當中有「yenjinc.info」這幾個字串，那麼就需要使用「grep」這個工具來搜尋了！接下來，我們就要來簡單認識一下 grep 工具的使用方法囉！

```
$ man grep

GREP(1)                                                    GREP(1)

NAME
    grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS
    grep      [options]      PATTERN      [FILE...]
```

「grep 用法」

```
$ grep [參數] 相關字串 [檔案]
$ grep          "yenjinc.info" /
$ grep  -n      "yenlung"      /etc/passwd

cyj@riemann:~$ grep cyj /etc/passwd
cyj:x:1000:1000:,,,:/home/cyj:/bin/bash
(直接找尋 /etc/passwd 中有關 cyj 的字串)

cyj@riemann:~$ cat /etc/passwd | grep cyj
cyj:x:1000:1000:,,,:/home/cyj:/bin/bash
(與上一個例子是一樣的，不過這裡是採用將 grep 接在 Pipe line 之後，這是常見的使用方式)

cyj@riemann:~$ grep -n yenlung /etc/passwd
25:yenlung:x:1003:1003:,,,:/home/yenlung:/bin/bash
(加上「-n」將找到的行數內容也標示出來，表示這次搜尋結果，yenlung 出現在檔案的第 25 行)

cyj@riemann:~$ grep -n false /etc/passwd
19:Debian-exim:x:102:102:./var/spool/exim4:/bin/false
21:identd:x:100:65534:./var/run/identd:/bin/false
```

```
22:sshd:x:101:65534::/var/run/sshd:/bin/false
```

(同樣的一個例子，將 `/etc/passwd` 中，含有 `false` 字串內容的行數都列出來)

```
cyj@riemann:~$ grep -v sh /etc/passwd
```

```
sync:x:4:65534:sync:/bin:/bin/sync
```

```
Debian-exim:x:102:102::/var/spool/exim4:/bin/false
```

```
identd:x:100:65534::/var/run/identd:/bin/false
```

(「`-v`」表示「反向搜尋」的意思！所以這個例子表示，將 `/etc/passwd` 檔案中，所有包含 `sh` 的內容都「不要」顯示出來！換句話說，也就是將「不包含」`sh` 的內容都要顯示出來。)

「grep」一些其他常用的方式

```
PowerBook-G4-12:- Denny$ history | grep ssh |grep cyj
```

```
78 ssh cyj@140.119.175.XX
```

```
228 ssh cyj@140.119.175.XX
```

```
262 ssh cyj@140.119.175.XX
```

```
347 ssh cyj@140.119.175.XX
```

```
349 ssh cyj@140.119.175.XX
```

(使用「`grep`」以及「導管」來過濾一些程式輸出的內容。這一個範例即是將所有「`history`」中，包含 `ssh` 以及 `cyj` 的結果都顯示出來！)

```
yenjinc@giant:~/Products$ find . -type f -exec grep -Hn "plone" {} \;
```

```
./skins/sin/sinList.pt:5: i18n:domain="plone">
```

```
./skins/sin/sinBox.pt:3: i18n:domain="plone">
```

```
./skins/sin/sincfg.pt:5: i18n:domain="plone">
```

```
./skins/sin/rss_slot.pt:5: i18n:domain="plone">
```

(使用 `find` 在現行目錄中找尋所有檔案，並將這些檔案內容有包含「`plone`」字串的「檔案名稱」跟「行數」標示出來！)

```
cyj@riemann:~$ cat /boot/grub/menu.lst |grep -v ^# |grep -v ^$
```

```
default 0
```

```
timeout 5
```

```
color cyan/blue white/blue
```

```
title Debian GNU/Linux, kernel 2.4.27-2-386
```

```
root (hd0,4)
```

```
kernel /vmlinuz-2.4.27-2-386 root=/dev/hda1 ro
```

```
initrd /initrd.img-2.4.27-2-386
```

```
savedefault
```

```
boot
```

```
title Debian GNU/Linux, kernel 2.4.27-2-386 (recovery mode)
```

```

root          (hd0,4)
kernel        /vmlinuz-2.4.27-2-386 root=/dev/hda1 ro single
initrd        /initrd.img-2.4.27-2-386
savedefault
boot

```

這一個例子，簡單來說即是將 `/boot/grub/menu.lst` 檔案，把所有「以 # 為開頭」以及「空白行」統統拿掉之後的結果顯示出來，這個例子使用了一些「正規表示式」，筆者會在下一節做介紹。

4.6 萬用字元與正規表示式

老實說！在這本書上講萬用字元與正規表示式是蠻狠的，不過這基本功如果打好的話，以後在使用或管理你的 Linux 系統時，一定會更方便好用，所以呢！筆者在下面將會有詳細的解說與實做。希望讀者可以吸收，如果看不懂也沒關係，等你用 Linux 久了以後，再來看看這個章節一定也會有所幫助的。

4.6.1 萬用字元

在系統的 shell 底下，擁有一些萬用字元 (wild card)，這些萬用字元對於檔案的搜尋相當好用哦！當然啦！萬用字元除了在檔案搜尋可以使用以外，平常各位在處理一些事情的時候，若使用萬用字元也會變得相當方便！下面簡單舉例萬用字元的使用方法。

符號	說明
?	萬用字元，代表此處一定要有「一個字元」
*	萬用字元，代表此處可以有「0 或多個字元」

尋找的字串	對映到的檔案
File?	File1 File2
File*	File File1

尋找的字串	對映到的檔案
	File1 a File1 b File2 File2a
?ile*	File file Lile File1 a File1 b File2 File2a

下面是萬用字元的一些基礎使用範例：

```

cyj@riemann:~$ ls
1.mp3  12.mp3  14.mp3  16.mp3  3.mp3  5.mp3  7.mp3  9.mp3      index.txt
11.mp3 13.mp3  15.mp3  2.mp3   4.mp3  6.mp3  8.mp3  Readme.txt install.txt
(目錄下有以下這些檔案)

cyj@riemann:~$ ls *.mp3
1.mp3  12.mp3  14.mp3  16.mp3  3.mp3  5.mp3  7.mp3  9.mp3
11.mp3 13.mp3  15.mp3  2.mp3   4.mp3  6.mp3  8.mp3
(將檔名為 mp3 的檔案全部列示出來)

cyj@riemann:~$ ls ?.mp3
1.mp3  2.mp3  3.mp3  4.mp3  5.mp3  6.mp3  7.mp3  8.mp3  9.mp3
(將「檔名只有一個字元」的 mp3 檔案全部列示出來)

cyj@riemann:~$ ls *.txt
Readme.txt  index.txt  install.txt
(將檔名為 txt 的檔案全部列示出來)

cyj@riemann:~$ ls i*.txt
index.txt  install.txt

```

(將所有「 i 開頭的 txt 檔案」全部列示出來)

```
cyj@riemann:~$ ls 1?.mp3
11.mp3 12.mp3 13.mp3 14.mp3 15.mp3 16.mp3
```

(將所有「十位數的 mp3 檔案」全部列示出來)

```
cyj@riemann:~$ ls 1*.mp3
1.mp3 11.mp3 12.mp3 13.mp3 14.mp3 15.mp3 16.mp3
```

(將所有「 1 開頭的 mp3 檔案」全部列示出來)

```
cyj@riemann:~$ ls ??????.txt
Readme.txt
```

(將檔案名稱「共有六個位元的 txt 檔案」全部列示出來)

```
cyj@riemann:~$ ls ?????.txt
index.txt
```

(將檔案名稱「共有五個位元的 txt 檔案」全部列示出來)

```
cyj@riemann:~$ ls ????.txt
ls: ????.txt: No such file or directory
```

(將檔案名稱「共有四個位元的 txt 檔案」全部列示出來，哇！原來此目錄下沒有檔案名稱為四個位元的 txt 檔案，所以 ls 找不到任何的結果。)

4.6.2 正規表示式

「正規表示式」在 Unix-like 系統當中，扮演著很重要的角色！詳細的用法與內容已經可以深入到一本書的容量了，不過在這裡，咱們暫時不看那些太過深入的細節，這邊只簡單介紹基礎的使用方法，詳細的內容就有待各位自己研究與摸索囉！以下是基礎常用的正規表示式與相關說明：

符號	說明	舉例
^	行的開頭	^The：搜尋開頭為 The 的內容
\$	行的結尾	End\$：搜尋結尾為 End 的內容
[abc]	符合集合中的字	[abc]：檔案中只要有 "a"、"b"、"c" 任一字即符合搜尋

符號	說明	舉例
[a-z]、[A-Z]	符合集合範圍中的字	[a-z]：符合 "a" 到 "z" 所有字元 [A-Z]：符合 "A" 到 "Z" 所有字元 [0-9]：符合 "0" 到 "9" 所有字元
.	任何單一字元	file.：符合 file1、file2，但不符合 file10
+	一個或多個字元	[0-9]+： 符合所有數字字元，表示檔案中只要有數字即符合搜尋
*	0 個或多個表示式	file.*：符合 file、file2、file10
?	0 個或一個表示式	file1?2：符合 file2、file12
	符合前面或者後面	file File：符合 file 也符合 File
綜合使用 (以身份證字號為例)		
[A-Z][12][0-9]{8}	P 1 33998337 R 2 12334567	(男性的身份證字號) (女性的身份證字號)

初學者對於「正規表示式」與「萬用字元」這兩者常常會搞混，因此正確的釐清觀念是很重要的喔！關於「正規表示式」的範例介紹，筆者覺得在《鳥哥的私房菜》中，這部份已經寫得相當完整囉！於是我不再重新撰寫這部份的範例介紹啦！

另外值得強調的是：建議各位可以跟著網頁的內容直接動手練習，這絕對有助於你更加熟悉正規表示式唷！

● 「鳥哥的私房菜-正規表示法」

http://linux.vbird.org/linux_basic/0330regularex.php

4.7 vi 編輯器

vi 是一個歷久彌新的編輯器，也是目前在 Unix-like 系統上被廣泛使用的一套編輯器，幾乎大部份的 Unix-like 系統上，各位都可以看到 vi 編輯器的存在，因此學好 vi 編輯器是一件很重要的事！當然啦！在 Unix-like 系統中，編輯器並不是只有 vi 一套而已，與 vi 相互抗衡的另一套強大編輯器叫做 emacs，雖然在這裡

介紹各位使用的是 vi 編輯器，不過筆者認為，多學一套編輯器是有利無害的，因為你總不能保證每台機器上都一定要有 vi 吧！:-P

4.7.1 vi 模式介紹

基本上 vi 共分為三種模式，分別是「一般模式」、「編輯模式」與「指令列命令模式」三種！

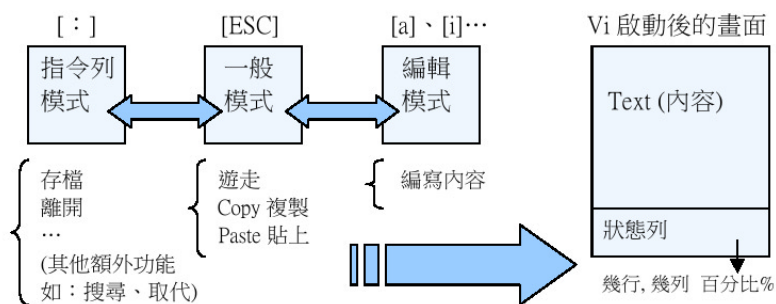


圖 4.12 vi 三種模式

為了讓大家可以更淺顯易懂的明瞭，簡單的說「編輯模式」就是各位在「key in (打字輸入資料)」的模式，而「一般模式」就是用來作些「複製、剪下、貼上...」甚至是「移動游標、遊走」的模式，最後的「指令列模式」其實就是用來「存檔、另存新檔...」相關的模式。當各位進入到 Vi 畫面之後，可以注意到畫面的最下方，有一個「狀態列」，此狀態列的右方，會即時顯示游標目前所在位置在「幾行幾列」以及目前游標位置所在檔案內容的「百分比」狀態。

4.7.2 vi 基本指令操作

有了基本的模式概念後，接著筆者將 Vi 常用的指令整理成表格給各位參考，假使各位手邊剛好有 Linux / Unix-like 機器，不妨一邊看書，一邊跟著實際練習看看，因為在這一章節裡，由於 Vi 本身是編輯器的關係，於是比較難以用「文字範例」來示範說明，所以「範例示範」就留給各位親自操刀囉！以下是常用的 Vi 指令整理列表：

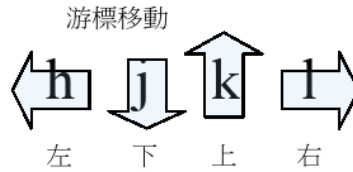


圖 4.13 vi 指標移動

編輯 (插入) 模式	
[i]	從「目前位置」開始編輯
[I]	從「行首」開始編輯
[a]	從「下個字元」開始編輯
[A]	從「行尾」開始編輯
[o]	產生「下一行」空白行
[O]	在「游標位置」建立空白行
[ESC]	切換回「一般模式」
一般模式	
[0]	到「行首」
[1G]	到「檔頭」
[\$]	到「行尾」
[G]	到「檔尾」
[w]	跳到「下一個字」
[x]	刪除「目前游標字元」
[X]	刪除「游標前一個字元」
[NUM] + [+]	往下「NUM」行
[NUM] + [—]	往上「NUM」行
[:] + [NUM]	跳到「NUM」行
[v]	選取範圍
[Ctrl] + [v]	「區塊」選取範圍
[y]	複製
[y][y]	複製一整行

[d]	刪除
[d][d]	刪除一整行
[d][0]	刪除至「行首」
[d][\$]	刪除至「行尾」
[d][1G]	刪除至「檔頭」
[d][G]	刪除至「檔尾」
[NUM] + [...] + [...]	例如：[3] + [d] + [d] 刪除三行
[...] + [NUM] + [...]	例如：[d] + [3] + [w] 刪除三個字
[x]	剪下 / 刪除
[p]	貼上
[u]	將字母「換成小寫」
[U]	將字母「換成大寫」
[~]	將字母「大小寫互換」
[r]	取代「一個」字元
[R]	「連續」取代
[u]	「復原」上一個動作
[Ctrl] + [r]	「取消復原」（重作的意思）
分頁編輯	
[Ctrl] + [w] + [n]	建立新分頁
[Ctrl] + [w] + [上(↑)]	切換到「上方分頁」
[Ctrl] + [w] + [下(↓)]	切換到「下方分頁」
[Ctrl] + [w] + [q]	關閉所在分頁
指令列模式	
[:][e]	編輯新檔
[:][w]	儲存檔案
[:][w][q]	儲存檔案後離開（也可以使用「:x」或「ZZ」）
[:][w][q][!]	「強制」儲存檔案後離開（加上！表示「強制」之意）
[:][q]	不儲存離開

[:][q][!]	「強制」不儲存離開
搜尋與取代	
[/]	向下搜尋 (N/n , N 切換至上一個, n 切換至下一個)
[?]	向上搜尋 (N/n , N 切換至上一個, n 切換至下一個)
:s/[old]/[new]/g	例如: s/Denny/Yenjin/g , 表示將所有的 Denny 替換為 Yenjin。
:1,30s/[old]/[new]/g	表示將「第一行」至「第三十行」的 [old] 替換成 [new]
:1,\$s/[old]/[new]/g	表示將整個檔案中的 [old] 替換成 [new]
:1,\$s/^/ /g	將所有檔案的行開頭都往後挪動四個空白 (用於排版用)
[Ctrl] + A	自動將數字累加

以上都是基礎常用的編輯方法，基本上熟悉這些用法後，應該就沒有什麼大問題啦！ :-D

4.8 帳號管理

4.8.1 背景知識

在 Unix-like/ Linux 系統當中，最重要的即是擁有「多人多工」的特性，也就是說，系統在開機之後，可同時讓許多人在上面作自己的工作。假使各位有使用過 Linux 或相關 Unix-like 的經驗，應該會曉得一台可以 login 的機器，它同時間是可以開放非常多人來使用的，也就是說每個使用者都可以同時間在這台機器上做自己的工作。而每位使用者在 login 這台機器時，是不是要輸入帳號與密碼呢？否則根本沒辦法登入的，換句話說，這台機器已經建立了每位使用者的帳號，於是所有人就可以使用自己的帳號跟密碼 login 進來使用囉！經過這樣的舉例說明，相信各位應該比較了解所謂「帳號」的這個概念了，接著下來，就要帶各位來認識「帳號管理」的一些相關指令使用，並舉例示範告訴大家，如何在系統中建立使用者帳號唷！

4.8.2 基本指令操作

在這裡會使用到的指令，筆者把它們整理列表如下：

建立帳號	
useradd	建立帳號 (手動)
adduser	建立帳號 (自動化)
更改密碼	
passwd	更改密碼
刪除帳號	
userdel	刪除帳號
相關設定檔	
/etc/passwd	帳號相關資訊
/etc/shadow	帳號密碼相關資訊

下面簡單舉例這些工具的使用方式：

手動建立帳號 (useradd)

```
# useradd [參數] [帳號名稱]
# useradd -m yenjinc
```

基本常用參數

```
-m
    The user's home directory will be created if it does not exist.
-p passwd
    The encrypted password, as returned by crypt(3). The default is
    to disable the account.
-s shell
    The name of the user's login shell.
```

(其他更多的參數，各位可以參照 manpage 唷！另外，有關帳號管理的所有指令，都必須擁有「超級使用者」root 的權限才行喔！)

```
debian:~# useradd -m -p "yenjinc.info" yenjinc
```


(建立一位「yenjinc」使用者，使用「-m」參數表示要建立「家目錄」，另外使用「-p」參數設定其密碼為「yenjinc.info」)

```
debian:~# ls -l /home | grep yenjinc
drwxr-xr-x  2 yenjinc  users      4096 Jul 25 14:04 yenjinc
```

(檢查一下，yenjinc 使用者已經被新增囉！)

```
debian:~# useradd -m -p "123" -s "/bin/sh" cyjtest
```

(建立一位「cyjtest」使用者，使用「-m」參數表示要建立「家目錄」，另外使用「-p」參數設定其密碼為「123」，使用「-s」參數設定其登入的 Shell 為「/bin/sh」)

```
debian:~# ls -l /home/|grep cyj
drwxr-xr-x  3 cyjtest      users              4096 Oct 18 21:38 cyjtest
```

```
debian:~# cat /etc/passwd|grep cyjtest
cyjtest:x:1036:100:::/home/cyjtest:/bin/sh
```

(驗證一下，已經成功建立「cyjtest」使用者，也順利建立「家目錄」以及更改「login Shell」)

自動化建立帳號 (adduser)

```
# adduser [帳號名稱]
# adduser  yenjinc
```

與 useradd 比較不同的是，adduser 會自動化幫你建立帳號相關資訊。

```
debian:~# adduser yenjinc
(使用「adduser」自動化建立使用者「yenjinc」)

Adding user `yenjinc'...
Adding new group `yenjinc' (1005).
Adding new user `yenjinc' (1005) with group `yenjinc'.
Creating home directory `/home/yenjinc'.
Copying files from `/etc/skel'
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for yenjinc
Enter the new value, or press ENTER for the default

    Full Name []:
    Room Number []:
    Work Phone []:
```

```

    Home Phone []:
    Other []:
Is the information correct? [y/N] y
(建立完成囉!)

更改密碼 (passwd)

# passwd [帳號名稱]
# passwd yenjinc

debian:~# passwd yenjinc
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
(密碼更改成功)

刪除帳號 (userdel)

# userdel [參數] [帳號名稱]
# userdel -r yenjinc

```

基本常用參數

```

-r      Files in the user's home directory will be removed along with
        the home directory itself and the user's mail spool.  Files
        located in other file systems will have to be searched for and
        deleted manually.

debian:~# userdel -r yenjinc
(刪除「yenjinc」使用者，由於加上「-r」參數，因此使用者「所有的相關資料」也都一併刪除！)

```

以上只是非常粗淺的使用範例而已，「帳號管理議題」絕對沒有像上面打打指令輕輕鬆鬆就解決了，其中還牽涉非常多的系統檔案內容與其他工具的使用！不過呢！以初學來講，咱們暫時還不需要了解的過於深入，初期的話，大概曉得工具該如何使用這樣就行囉！

4.9 行程管理

上一節筆者在介紹「帳號管理」時候，有簡單提到 Unix-like 系統多人多工的特性，在系統運行時，除了可以擁有很多人同時登入使用的特性以外，其實系統中是有很多的「工作程序」（Process）同一時間正在由「許多不同」的人正在執行當中呢！咱們只要透過一些「工具指令」就可以來查看目前系統中，有哪些正在運作的程序（Process），以及查看是哪些人正在使用它們哦！甚至還可以手動來「刪除」這些程序呢！接下來，筆者將要帶各位來認識這些基礎的行程管理工具使用方式。

4.9.1 前景與背景工作

當每位使用者 login 系統之後，所執行的任何一項工作都可視為一個「程序」（Process），藉由「Command Line」（文字終端機模式）各位可以立即的下達指令讓系統去執行它，而必須一直到這項程序「實行完畢」後，才能再下達另外一個指令給終端機，否則畫面就會一直「停」在那邊，這時候我們什麼事也不能做，只能等到它跑完，才能進行下一個動作。

譬如，筆者正在終端機下打包一個大資料時：

```
debian:~# tar -czvf usr.tgz /usr/
tar: Removing leading `/' from member names
```

這個時候畫面會整個「停」在那邊，我們只能等它跑完，否則什麼事情也不能做！於是在 Linux / Unix-like 系統底下，各位就可以利用一些「工作管理」指令來解決這類的問題囉！

筆者把這裡會使用到的使令，列整如下表：

暫停目前的工作程序	
[Ctrl] + z	按下之後會把目前的工作程序「暫停」
切換到背景執行	
bg	將目前工作程序丟到「背景」執行
&	與「bg」是一樣的（但使用方式不同）

切換到前景執行	
fg	將「背景工作」丟回「前景」執行
查看背景工作程序狀態	
jobs	「查看」背景工作程序「運行狀態」
刪除背景工作程序	
kill %n	「刪除」在背景執行的工作程序（「n」是工作的編號）

各位可以從下面這張簡圖了解一下「前景」與「背景」工作的相互關係：

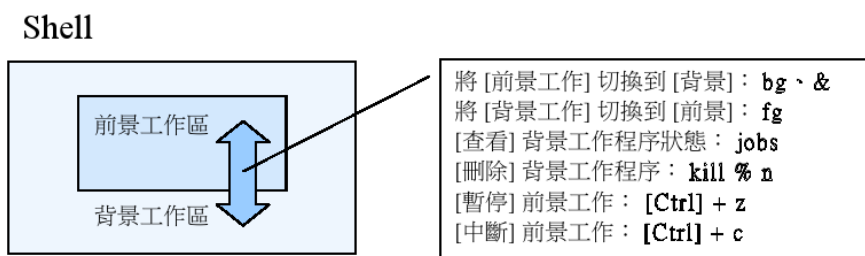


圖 4.14 前景與背景間的關係

下面簡單舉例示範這些工具的使用：

```

debian:~# tar -czf usr.tgz /usr/
(按下 [ctrl] + z，將目前工作「暫停」)
[1]+  Stopped                  tar -czf usr.tgz /usr/

debian:~# tar -czf usr2.tgz /usr/
(按下 [ctrl] + z，把第二個執行的工作「暫停」)
[2]+  Stopped                  tar -czf usr2.tgz /usr/

debian:~# jobs
(使用「jobs」指令，查看目前背景工作的程序狀態，可以看到有兩個工作被「暫停」了)
[1]-  Stopped                  tar -czf usr.tgz /usr/
[2]+  Stopped                  tar -czf usr2.tgz /usr/

debian:~# bg 1
(使用「bg」指令，將「工作程序 1」丟到背景執行)
[1]-  tar -czf usr.tgz /usr/ &

```

```

debian:~# tar -czf usr3.tgz /usr/ &
(建立第三個工作程序，並使用「&」符號，將它直接丟到背景執行)
[3] 6042

debian:~# jobs
(使用「jobs」指令，查看目前背景工作的程序狀態，可以看到第一個與第三個工作程序正在「Running」，
而第二個工作依然被「暫停」住)
[1]  Running                tar -czf usr.tgz /usr/ &
[2]+  Stopped                tar -czf usr2.tgz /usr/
[3]-  Running                tar -czf usr3.tgz /usr/ &

debian:~# kill %3
(使用「kill」指令，「刪除」第三個背景工作程序)

debian:~# jobs
(使用「jobs」指令，在查看一次目前背景工作的程序狀態，可以看到「工作 3」已經「終止」了)
[1]  Running                tar -czf usr.tgz /usr/ &
[2]+  Stopped                tar -czf usr2.tgz /usr/
[3]-  Terminated           tar -czf usr3.tgz /usr/

debian:~# kill %1
(使用「kill」指令，「刪除」第一個背景工作程序)

debian:~# jobs
(使用「jobs」指令，在查看一次目前背景工作的程序狀態，可以看到「工作 1」也被「終止」了)
[1]-  Terminated           tar -czf usr.tgz /usr/
[2]+  Stopped                tar -czf usr2.tgz /usr/

debian:~# fg 2
(使用「fg」指令，將「工作程序 2」丟回前景執行)
tar -czf usr2.tgz /usr/

```

4.9.2 基本指令操作

上一節是簡單的舉例「工作管理」的使用方法，接下來要介紹各位的就是比較深入一點的「程序管理」啦！程序管理的內容就會比較多喔！不過初步來講，各

位先有個基本的概念就行囉！所以下面筆者將要初淺的介紹一些基本「程序管理」指令的使用方式。

這裡會用到的指令，筆者將它們列整如下：

查看系統程序	
ps	查看系統程序
pstree	查看系統程序 (以「樹狀圖」顯示)
系統資源觀察	
top	「即時性」的查看系統資源狀態
刪除背景工作程序	
kill PID	刪除系統程序 (PID 是程序的編號)
killall	刪除以某程序為名的所有相關程序

我們來看一些基本的操作範例：

使用「ps」工具，列出系統「所有正在執行」的程序

```
ps 常用參數

-a      Display information about other users' processes as well as your own.
        (顯示所有附屬於「終端機」的行程)

-u      Display information associated with the following keywords: user, pid,
        %cpu, %mem, vsz, rss, tt, state, start, time and command.
        (以「使用者格式」與其他相關長格式顯示)

-x      Display information about processes without controlling terminals.
        (顯示所有「不是由終端機」所控制的行程)

debian:~# ps -ax
  PID TTY          STAT       TIME COMMAND
    1 ?            S          0:03 init [2]
    2 ?            S          0:00 [keventd]
    3 ?            SN         0:00 [ksoftirqd_CPU0]
    5 ?            S          0:00 [bdflush]
.....略.....
```

✎ 使用「ps」加上「-u」參數，列出包含「使用者名稱」與「更完整資訊」的所有正在執行的程序

```

debian:~# ps -aux
USER      PID  %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.0  0.2  1496   508 ?        S    Jul21   0:03 init [2]
root         2   0.0  0.0     0     0 ?        S    Jul21   0:00 [keventd]
root         3   0.0  0.0     0     0 ?        SN   Jul21   0:00 [ksoftirqd_CPU0]
root         4   0.0  0.0     0     0 ?        S    Jul21   0:00 [kswapd]
root         5   0.0  0.0     0     0 ?        S    Jul21   0:00 [bdflush]
root         6   0.0  0.0     0     0 ?        S    Jul21   0:00 [kupdated]
root        99   0.0  0.0     0     0 ?        S    Jul21   0:04 [kjournald]
root       1261   0.0  0.2  1544   620 ?        Ss   Jul21   0:02 /sbin/syslogd
root       1264   0.0  0.5  2176  1248 ?        Ss   Jul21   0:00 /sbin/klogd
.....略.....

```

✎ 使用「pstree」工具，將系統正在執行的程序以「樹狀圖」來顯示

```

debian:~# pstree
init--|--atd
      |--bdflush
      |--cron
      |--exim4
      |--6*[getty]
      |--inetd
      |--keventd
      |--khubd
      |--kjournald
      |--klogd
      |--ksoftirqd_CPU0
      |--kswapd
      |--kupdated
      |--lpd
      |--portmap
      |--rpc.statd
      |--sshd--|--sshd--|--sshd
      |
      |   |--sshd--|--bash--|--bash--|--pstree
      |
      |--syslogd

```

使用「top」工具，即時性的查看系統資源狀態

```

debian:~# top
top - 14:50:21 up 5 days, 3:09, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 30 total, 1 running, 29 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0% user, 0.0% system, 0.0% nice, 100.0% idle
Mem: 248980k total, 214164k used, 34816k free, 16200k buffers
Swap: 738948k total, 0k used, 738948k free, 161784k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        8   0   508   508  448  S   0.0   0.2   0:03.90  init
    2 root        9   0     0     0    0  S   0.0   0.0   0:00.03  keventd
    3 root       19  19     0     0    0  S   0.0   0.0   0:00.00  ksoftirqd_CPU0
    4 root        9   0     0     0    0  S   0.0   0.0   0:00.18  kswapd
.....略.....

```

刪除系統執行中的程序

```

debian:~# tar -zcf usr1.tgz /usr/ &
[1] 6484
debian:~# tar -zcf usr2.tgz /usr/ &
[2] 6486
debian:~# tar -zcf usr3.tgz /usr/ &
[3] 6488
(臨時建立三個運作程序，並將它們全部丟到背景執行)

debian:~# ps -uax | grep tar
(將「ps -aux」的顯示結果，透過「grep」來過濾出那些臨時建立的「tar 程序」)
root      6484  2.0  0.3  2232  860 pts/0    S   14:56   0:00  tar zcf usr1.tgz /usr/
root      6486  1.5  0.3  2232  852 pts/0    S   14:56   0:00  tar zcf usr2.tgz /usr/
root      6488  0.8  0.3  2232  852 pts/0    S   14:56   0:00  tar zcf usr3.tgz /usr/
(從這裡可以看到每一個工作行程都有一組數字 ID，稱作為「PID」。例如，這三個 tar 程序的 PID 分別是「6484」、「6486」、「6488」)

debian:~# kill 6484
(使用「kill」指令，將PID為「6484」的工作程序刪除！)

debian:~# ps -uax | grep tar

```



```
(透過「grep」來過濾出「tar 程序」，發現原先的「6484」已經刪除囉！)
root      6486  1.5  0.3  2232  860  pts/0    S    14:56   0:00  tar zcf usr2.tgz
/usr/
root      6488  1.5  0.3  2232  860  pts/0    S    14:56   0:00  tar zcf usr3.tgz
/usr/
[1] Terminated          tar zcf usr1.tgz /usr/
(由於 6484 程序是在背景執行的，所以刪除之後，前景畫面會顯示「工作 1 終止」的提示)

debian:~# killall tar
(接著使用「killall」指令，將「所有的 tar 工作程序」全部刪除！)
[2]- Terminated          tar zcf usr2.tgz /usr/
[3]+ Terminated          tar zcf usr3.tgz /usr/
debian:~# ps -uax | grep tar
(檢查一下！會發現所有的「tar」程序已經都不見囉！而且前景畫面也出現了工作 2、工作 3「終止」的提示！)
```

以上這些就是「程序管理」中，最基礎的使用範例介紹！

CHAPTER 5

套件管理系統

在 GNU/Linux 剛開始發展的初期，安裝軟體都要自己抓原始碼（通常以 tarball 的型式），解壓縮，設定後經過編譯，再手動安裝。編譯的時候還要注意編譯時所需要的函式庫和表頭檔，並且編譯所需的時間很久，這也造成初學者很大的困擾和門檻。

經過那個時代或玩過嵌入式系統的人，都會對以下指令很懷念吧！

```
./configure  
make  
make install
```

讀者應該記得常會在 configure 的時候，缺一堆函式庫或表頭檔，然後還要自己把缺的慢慢補上去。等到全部弄好時，都已經過了很久的時間，耗時耗力，尤其有急用的時候，這樣真的太不切實際了。

備註 在嵌入式系統中，可能不提供套件管理系統，或者只提供功能受限的替代品，在建構系統的過程中，需要考慮到函式庫相依性問題與特定資源使用的需求，過程繁瑣。而編譯這些軟體通常會從 tarball 安裝。大部份自由軟體釋出的原始碼會用 tar 打包成一個檔案，所以大家常常會把軟體原始碼套件稱為 tarball。

也因此有了套件管理系統的產生，例如 RadHat 公司所使用的 RPM 管理系統，或者 Debian 所使用的 DEB 套件格式。通常這種套件檔是二進位格式，也就是已經編譯好的，所以是只要安裝就可以用。不過這種套件檔還是有函式庫相依性的問題要注意，它只不過省略了編譯時所需的時間和表頭檔的相依性問題。為了解決這樣的問題，Debian 又引進了 APT 系統，它可以在你安裝軟體 A 的時候，自動幫你補上相依的其他套件。更強大的是，它甚至能幫你自動線上升級更新。

強大的 Debian 套件管理系統，被 Ubuntu 所繼承，因此所有 Debian 系統的觀念，幾乎都可以在 Ubuntu 上使用唷！

這裡筆者要大家特別留意的是，早期很多的 GNU/Linux 的書籍教大家伺服器相關軟體最好手動從最新的 tarball 編譯，以避免一些安全性上的問題。但這在近期新的 Linux 發行版上，這種觀念是錯誤的唷！因為早期的發行版沒有類似 APT 系統，可以自動上網更新，所以乾脆自己安裝最新版，或許會比較安全。但你有看過誰閒閒沒事每天看漏洞公告？這種瑣事，有人代勞幫忙看漏洞公告，並且由他們提供最新的漏洞修補，這樣不是輕鬆多了嘛！因此我非常不建議使用者伺服器的相關套件從 tarball 自己編譯來裝，因為這樣哪天有漏洞時，使用者還要自己弄補釘。這還不打緊，怕的是連漏洞公告都沒有注意到。目前 Ubuntu 上桌面版的維護 3 年，伺服器版的維護 5 年。這段時間內有漏洞或臭蟲，將會支援自動更新，使用者不需要自行下載 patch 並將修正過的程式碼重新編譯。

5.1 DPKG 套件管理程式

DPKG 是 Debian 和 Ubuntu 用來管理套件的基礎程式，它可以用來安裝、解除安裝副檔名為 deb 的套件軟體，可以將它視為一種後端的管理工具，因為它只拿來做一些很底層的套件管理，無法做自動相依性問題的解決。但是 APT 系統其實底層跑的還是 DPKG，不過當遇到相依性問題，APT 系統會自動把相依的套件抓下來，再用 DPKG 把這些套件安裝好。所以我們一般會把 APT，Aptitude 和 synaptic 稱為前端管理系統，這些前端管理系統底層也是用 DPKG 來做套件的處理，不過它們可以順便幫我們處理一些安裝的瑣事，例如升級和相依性問題！我們可以透過圖 5.1 來了解整個套件管理系統的架構。

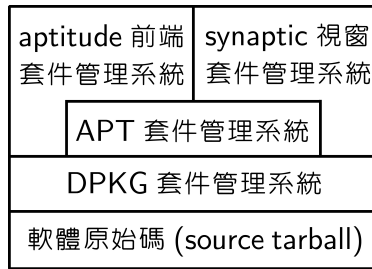



圖 5.1 套件管理系統架構

你會發現，在執行這些前端管理程式時，若下載完軟體，當實際安裝的時候，dpkg 這個程式會在背景執行唷！你可以用以下指令來驗證呢！

 指令：

```
ps -aux | grep dpkg
```

5.1.1 deb 套件檔格式解說

Ubuntu 的套件檔裡面，可能是可執行檔、函式庫和文件等，而文件套件可以在各個平台上共用，所以是不分平台的，基本上這些套件檔都是以 deb 來做副檔名喔！舉例來說，套件都是用以下規則來命名：

```
foo_ver-rev_arch.deb
```

foo 代表了套件名稱，ver 是套件版本，也就是原作者釋出時所給與的編號。rev 是代表 Ubuntu 的第幾次釋出，通常是 Ubuntu 做細部修改後，才會以 rev 來做區別。至於 arch 就是分別代表著適用在不同平台上的版本，有可能是 i386，amd64 和 powerpc 等。但如果 arch 是 all 的話，代表這個套件可以在不同平台上使用，所以這些套件通常是文件或者是用直譯語言寫的程式，不需經過編譯。


或許這樣講還是有點籠統，筆者就舉個實際的例子來讓大家比較容易了解好了，這裡的例子是有名的網頁伺服器，Apache。為了讓使用者可以決定哪些功能是否需要用到，Ubuntu 把 Apache 拆成以下多個套件。

套件檔案名稱	說明
apache2-doc_2.0.55-4ubuntu2_all.deb	這是 Apache 的說明文件。注意，在 arch 的那個欄位，它是 all，所以這個套件是不分平台的喔！
apache2_2.0.55-4ubuntu2_i386.deb	這是一個虛擬套件，安裝這個套件，它就會利用相依性的關係自動安裝下面 3 個套件啦！
apache2-common_2.0.55-4ubuntu2_i386.deb	這裡面包含了一些 Apache 所使用的模組。
apache2-mpm-worker_2.0.55-4ubuntu2_i386.deb	這是 Apache 實際的執行檔喔！
apache2-utils_2.0.55-4ubuntu2_i386.deb	這包含了一些 Apache 所使用的工具程式。

看完了上面的表格，讀者大概對於套件檔命名有初步了解了吧！在套件的檔名中，是使用下底線（_）來分隔出 3 個區塊。如 apache2-common_2.0.55-4ubuntu2_i386.deb，它的第一個區塊就是 apache2-common，也就是套件名稱。而第二個區塊是 2.0.55-4ubuntu2，也就是本版號加上 rev 號，在套件系統中，比對軟體新舊是用這個區塊直接比對，遇到英文的部份就以它 ASCII 編號排序，也就是 ver 和 rev 是放在一起比較。最後才是平台版本。

5.1.2 DPKG 查詢套件

在 Ubuntu 中，我們常常使用以下指令來查詢某個已安裝套件的資訊，

 指令：

```
dpkg -l <套件名稱> # 查詢套件的資訊
```

 範例：

```
dbtsai@ubuntu:/$ dpkg -l filezilla
期望狀態=未知(u)/安裝(i)/刪除(r)/清除(p)/保持(h)
| 當前狀態=未(n)/已安裝(i)/僅存設定(c)/僅解壓縮(U)/設定失敗(F)/不完全安裝(H)
|/ 錯誤?=(無)/保持(?)/須重裝(R)/兩者兼有(+) (狀態，錯誤：大寫=故障)
```

```

||/ 名稱          版本          簡介
+++-----
ii filezilla      3.0CVS20060622 GUI FTP client written in wxWidgets
dbtsai@ubuntu:/$

```

在此範例中左下角 ii 的代碼是代表該套件在系統上的狀態。它的第一碼稱為期望狀態，也就是我們對套件狀態的期望，例如安裝，刪除或者清除等。但事情有時候總是不如人意的，所以第二碼就是所謂的套件目前狀態，也就是你希望套件的狀態在某個情形，但實際狀態是如第二碼所示。表 5.1 和表 5.2 分別列出了期望狀態和目前狀態的代碼解釋。


表 5.1 期望狀態代碼

期望狀態代碼	說明
u	未知 (Unknown)
i	安裝 (Install)
r	刪除 (Remove)
p	清除 (Purge)
h	凍結 (Hold)


表 5.2 目前狀態代碼

目前狀態代碼	說明
n	無 (Not)
i	已安裝 (Installed)
c	僅存設定檔 (Config-files)
U	僅解壓縮 (Unpacked)
F	設定失敗 (Failed-config)
H	不完全安裝 (Half-installed)

例如我們常使用以下兩個幾乎同等意義（差別在自動解決相依性）的指令來移除套件。

 指令：


```
apt-get remove <套件名稱> # 刪除套件但不刪除設定檔 (會自動解決相依性)
```

 指令：

```
dpkg -r <套件名稱> # 刪除套件但不刪除設定檔 (不會自動解決相依性)
```

這兩個指令，系統預設是把將移除套件的設定檔留下來，只有把程式移除。會不移除設定檔是因為有時候是程式出了問題，我們想要把它刪除後再重新安裝，如果移除套件時，系統也把設定檔砍了，這樣對管理者很不方便！畢竟那些設定檔要費很多苦工才能設定好，所以 Ubuntu 在移除套件的時候，預設是把設定檔留下來的。如以下範例，筆者用 `apt-get remove postfix` 以後，再去查詢套件資訊，就會有 `rc` 的狀態碼，表示套件已經被移除，但是設定檔還留在系統。如果讀者需要連設定檔都移除，就必須使用 `dpkg -P postfix` 來移除設定檔，或者是在還沒移除前，使用 `apt-get --purge remove postfix` 來移除！


當套件已經被移除了，也就是狀態碼已經是 `rc` 的時候，`apt-get --purge` 這個方式沒辦法砍設定檔，這個方式只能在還沒移除前使用來反安裝程式和移除設定檔。若要在 `rc` 狀態的時候砍設定檔，要使用 `dpkg -P` 的方式！

 範例：


```
dbtsai@ubuntu:~$ sudo apt-get remove postfix
下列的套件都將被【刪除】：
    postfix
更新 0 個套件，新安裝 0 個套件，刪除 1 個套件，另不更新 0 個套件。
繼續執行嗎？ 是按 [Y] 鍵，否按 [n] 鍵 y
正在刪除 postfix ...

dbtsai@ubuntu:~$ dpkg -l | grep postfix
rc postfix                2.2.10-1ubuntu0.1      A
high-performance mail transport agent
dbtsai@ubuntu:~$ sudo dpkg -P postfix
(正在讀取資料庫 ... 系統目前總共安裝有 171104 個檔案和目錄。)
正在刪除 postfix ...
正在清除 postfix 的設定檔案 ...
dbtsai@ubuntu:~$ dpkg -l | grep postfix
dbtsai@ubuntu:~$
```


當你不知道套件完整名稱時，可以使用以下指令透過關鍵字去查詢套件：

 指令：


```
dpkg -l | grep <套件關鍵字> # 查詢包含關鍵字的套件資訊
```

 範例：


```
dbtsai@ubuntu:/$ dpkg -l | grep filezilla
ii filezilla 3.0CVS20060622-3 GUI FTP client written in wxWidgets
dbtsai@ubuntu:/$
```

而它的原理就是 `dpkg -l` 這個指令會列出所有安裝的套件，因此可以透過管線把資訊傳給 `grep` 去抓取關鍵字，如以上範例所示。

若是你想要知道某個套件包含哪些檔案的話，可以透過下面指令：


 指令：

```
dpkg -L <套件名稱> # 查詢套件包含的所有檔案
```


 範例：

```
dbtsai@ubuntu:/$ dpkg -L filezilla
/usr/bin/filezilla
/usr/bin/fzsftp
以下略
dbtsai@ubuntu:/$
```

若是要列出某個 `deb` 套件檔包含的檔案內容，可以透過這個指令，基本上它和上面那個指令是類似的，不過一個是查詢已安裝套件，這個是針對 `deb` 套件檔。

 指令：

```
dpkg -c <.deb 檔案名稱> # 查詢 deb 套件檔包含的所有檔案
```


 範例：

```
dbtsai@ubuntu:~/deb$ dpkg -c filezilla_3.0CVS20060622_powerpc.deb
-rwxr-xr-x root/root 1288788 2006-07-08 16:00:43 ./usr/bin/filezilla
-rwxr-xr-x root/root 379384 2006-07-08 16:00:43 ./usr/bin/fzsftp
```


以下略

```
dbtsai@ubuntu:/$
```

若是想要得知某個檔案是哪個套件所提供的，可以透過下面指令：


 指令：

```
dpkg -S <預查詢的檔案名稱> # 查詢檔案是哪個套件所提供的
```


 範例：

```
dbtsai@ubuntu:~$ dpkg -S /usr/bin/fzsftp
filezilla: /usr/bin/fzsftp
dbtsai@ubuntu:~$
```

若想要查詢已經安裝的套件詳細資訊，可以透過下面指令：


 指令：

```
dpkg -s <套件名稱> # 查詢該已經安裝套件的詳細資訊
```


 範例：

```
dbtsai@ubuntu:~/deb$ dpkg -s filezilla
Package: filezilla
Status: install ok installed
Priority: optional
Section: x11
Installed-Size: 3319
Maintainer: Tsai Dung-Bang <dbtsai@gmail.com>
Architecture: powerpc
Version: 3.0CVS20060622
Depends: libc6 (>= 2.3.4-1), libexpat1 (>= 1.95.8), libgcc1 (>= 1:4.0.2), libidn11 (>= 0.5.18), libstdc++6 (>= 4.0.2-4)
Description: GUI FTP client written in wxWidgets
 FileZilla is a fast and reliable FTP client, with lots of useful
 features and an intuitive interface.
以下略
dbtsai@ubuntu:/$
```

若你要了解 deb 套件檔的詳細資訊，可以用下面指令，它的輸出資訊幾乎和上面那個一樣，只不過一個是針對已經安裝的套件，這個是針對套件檔喔！

 指令：


```
dpkg -I <.deb 檔案名稱> # 查詢該 deb 套件檔的詳細資訊
```

 範例：


```
dbtsai@ubuntu:~/deb$ dpkg -I filezilla_3.0CVS20060622_powerpc.deb
Package: filezilla
Version: 3.0CVS20060622
Section: x11
Priority: optional
Architecture: powerpc
Depends: libc6 (>= 2.3.4-1), libexpat1 (>= 1.95.8), libgcc1 (>= 1:4.0.2),
libidn11
Installed-Size: 3319
Maintainer: Tsai Dung-Bang <dbtsai@gmail.com>
Description: GUI FTP client written in wxWidgets
 FileZilla is a fast and reliable FTP client
以下略
dbtsai@ubuntu:/$
```

5.1.3 DPKG 安裝套件

若要安裝一個套件檔，我們在 Ubuntu 會使用以下指令：

 指令：

```
sudo dpkg -i <.deb 檔案名稱> #安裝套件
```

 範例：

```
dbtsai@ubuntu:~/deb$ sudo dpkg -i filezilla_3.0CVS20060622_powerpc.deb
選中了曾被取消選擇的套件 filezilla。
正在解壓縮 filezilla (從 filezilla_3.0CVS20060622_powerpc.deb) ...
正在設定 filezilla (3.0CVS20060622) ...
dbtsai@ubuntu:~/deb$
```

安裝時，若有失敗的情形出現，可能是因為和現有套件有衝突（conflict）或者是無法滿足套件的相依性（dependency）。

衝突的問題通常是某些套件不能同時安裝，也許因為它們有相同功能所以不能裝在一起等原因。這時候只要把互相衝突的軟體反安裝，應該就可以解決了。

若是相依性問題，如以下範例，`apache2_2.0.55-4ubuntu2_powerpc.deb` 相依於 `apache2-mpm-worker`，`apache2-mpm-prefork` 或者 `apache2-mpm-perchild` 三者的其中一個。因此安裝的時候就得要先安裝這三個的其中一個。這裡 Apache 有三種版本的執行檔是因所採用的多工執行模式不同，不過這不在我們現在的討論範圍，詳情會在 14.2.1 小節解釋。

範例：


```
dbtsai@ubuntu:~/apache2$ sudo dpkg -i apache2_2.0.55-4ubuntu2_powerpc.deb
dpkg: 相依關係問題使得 apache2 的設定工作不能繼續：
  apache2 相依於 apache2-mpm-worker (= 2.0.55-4ubuntu2) | apache2-mpm-prefork (= 2.0.55-4ubuntu2) | apache2-mpm-perchild (= 2.0.55-4ubuntu2); 然而：
    套件 apache2-mpm-worker 還沒有被安裝。
    套件 apache2-mpm-prefork 還沒有被安裝。
    套件 apache2-mpm-perchild 還沒有被安裝。
dpkg: 處理 apache2 (--install)時出錯：
  相依關係問題 - 仍未被設定
在處理時有錯誤發生：
  apache2
dbtsai@ubuntu:~/apache2$
```

因此我們必須先安裝 `apache2-mpm-worker`，但如果你試過的話，會發現 `apache2-mpm-worker` 又相依於 `apache2-common`，而 `apache2-common` 相依於 `apache2-utils`。所以這時候軟體的安裝順序就要是先裝 `apache2-utils`，然後再裝 `apache2-common`，之後再裝 `apache2-mpm-worker`，最後安裝 `apache2`。當然也可以不用那麼麻煩，可以把所有相關的檔案放在同一個目錄，然後透過以下指令一次把所有 deb 檔安裝，它順便也會處理安裝順序。

指令：

```
sudo dpkg -i *.deb # 安裝目錄下所有 deb 套件檔
```

當然最方便的方式是透過 apt 系統安裝，這樣我們連把這些套件套載到硬碟都不用，只要以下一個指令就全部都設定好啦！

 指令：

```
sudo apt-get install apache2 # 安裝 Apache 網頁伺服器
```

在安裝或解除安裝的時候，若有遇到一些錯誤訊息或相依性而導致的失敗，有時候可以強制把它裝上去或反安裝。但有時候，這種蠻力法常常會造成更多問題，所以非必要的話，這種方式筆者不建議使用。此法有可能會破壞系統相依性，然後在下次系統 `apt-get upgrade` 的時候，系統可能會想要試著去修復，但是因為先前的強制安裝，導致系統自動把強制安裝的那些套件反安裝，來滿足系統的相依性。

使用方式為原來的安裝指令加上強制操作參數，而參數可以有好幾個，用逗點隔開。例如可以用參數 `--force-hold,bad-verify` 來達到可以對被鎖定且沒通過驗證的 `deb` 套件做安裝。而表 5.2 列出了一些常用的參數，讀者可以透過 `dpkg --force-help` 來了解其他的參數。

表 5.2 DPKG 強制操作參數

出現警告時，根據參數強迫執行： <code>--force-<參數> , <參數> , ...</code>	
出現警告時，根據參數強迫停止： <code>--refuse-<參數> , <參數> , ...</code>	
參數	說明
<code>all</code>	應用所有下列的強制方法。
<code>hold</code>	對被標記為保持原狀 (<code>hold</code>) 的套件進行操作。
<code>not-root</code>	即使使用者不是 <code>root</code> ，也試圖安裝或移除軟體。
<code>overwrite</code>	使用來自於安裝套件的檔案覆蓋到原本屬於另一個套件的檔案。
<code>bad-verify</code>	即使套件沒通過驗證，也將會被安裝。
<code>depends-version</code>	把所有因版本而造成的相依性問題略過並繼續安裝 (可能造成系統損壞)。
<code>depends</code>	把所有類型的相依性問題略過並繼續安裝 (可能造成系統損壞)。
<code>confnew</code>	總是使用新版設定檔案，並不再詢問 (可能造成系統損壞)。
<code>confold</code>	總是使用原有的設定檔案，並不再詢問 (可能造成系統損壞)。
<code>conflicts</code>	讓相互衝突的套件也能安裝 (可能造成系統損壞)。


出現警告時，根據參數強迫執行： <code>--force-<參數> , <參數> , ...</code>	
出現警告時，根據參數強迫停止： <code>--refuse-<參數> , <參數> , ...</code>	
參數	說明
<code>overwrite-dir</code>	用來自一個套件的檔案覆蓋原本屬於另一套件的目錄（可能造成系統損壞）。
<code>remove-reinstreq</code>	刪除那些一定需要安裝的套件（可能造成系統損壞）。
<code>remove-essential</code>	刪除標記為重要的套件（可能造成系統損壞）。

例如筆者抓到一個 deb 套件檔，它是在比筆者還要新的編譯環境下所編譯，因此如範例所示，有許多的版本相依問題出現，造成不能安裝。

範例：


```
dbtsai@ubuntu:~/deb$ sudo dpkg -i mplayer_0.99+1.0pre8-0ubuntu4_powerpc.deb
正預備替換 mplayer 2:0.99+1.0pre8-0ubuntu4 (使用
mplayer_0.99+1.0pre8-0ubuntu4_powerpc.deb) ...
正在解壓縮替換的套件檔 mplayer ...
dpkg: 相依關係問題使得 mplayer 的設定工作不能繼續:
 mplayer 相依於 libasound2 (> 1.0.11); 然而:
 系統裡的 libasound2 的版本是 1.0.10-2ubuntu4.
 mplayer 相依於 libatk1.0-0 (>= 1.12.1); 然而:
 系統裡的 libatk1.0-0 的版本是 1.11.4-0ubuntu1.
 mplayer 相依於 libc6 (>= 2.4-1); 然而:
 系統裡的 libc6 的版本是 2.3.6-0ubuntu20.
以下略
dbtsai@ubuntu:~/apache2$
```

此時可以透過以下指令來強制安裝，這個時候會出現很多的錯誤訊息。雖然是安裝成功了，並且軟體有一定的機率可以正常使用（有時候硬裝的套件和系統環境差異太大時，還是不能使用喔！）。但這樣在下次系統更新的時候，APT 系統為了自動恢復系統相依性的一致性，會把這樣惡搞的套件刪除。

 指令：


```
sudo dpkg -i --force-all mplayer_0.99+1.0pre8-0ubuntu4_powerpc.deb
```

如果確定經過惡搞後可以使用，為了不讓系統在升級的時候把惡搞版的套件移除，可以用以下方式來修改套件內環境相依版本資訊。

 指令：


```
ar x mplayer_0.99+1.0pre8-0ubuntu4_powerpc.deb
```

把套件檔解開，你會發現有 `debian-binary`，`control.tar.gz` 和 `data.tar.gz` 3 個檔案。其中 `control.tar.gz` 負責紀錄套件的一些資訊。然後再把 `control.tar.gz` 解開到臨時的目錄，你會發現有一個 `control` 的文字檔。你可以把 `Depends` 那個欄位改成你要的，例如可以把版本號改小，或是把它全砍了。然後再原封不動的壓縮成 `control.tar.gz`，再用

 指令：

```
ar d mplayer_0.99+1.0pre8-0ubuntu4_powerpc.deb control.tar.gz
```

把原來的 `control.tar.gz` 從 `deb` 檔砍了，最後用

 指令：

```
ar r mplayer_0.99+1.0pre8-0ubuntu4_powerpc.deb control.tar.gz
```

把惡搞版的紀錄檔加回去。這時候你會發現，新的套件檔已經沒有相依性的問題喔！

備註 其實這是用修改套件紀錄的方式去騙過 DPKG 套件管理系統。

5.1.4 DPKG 移除套件

移除套件有兩種指令，一種是只移除主程式，這時候該套件的設定檔會留在系統內，此時該套件的狀態碼就會是 `rc` 了！另外一種是連設定檔都一起移除。

第一種只移除套件的方式，結果如範例，套件狀態變成了 rc。

指令：

```
sudo dpkg -r <套件名稱> #移除套件但不移除設定檔
```

範例：

```
dbtsai@ubuntu:~/deb$ sudo dpkg -r filezilla
正在刪除 filezilla ...
dbtsai@ubuntu:~/deb$ dpkg -l | grep filezilla
rc filezilla 3.0CVS20060622 GUI
FTP client written in wxWidgets
dbtsai@ubuntu:~/deb$
```

第二種方式移除了程式和設定檔，你可以透過 `dpkg -l | grep <套件名稱>` 去查看，會發現已經找不到紀錄了唷！結果如下：

指令：

```
sudo dpkg -P <套件名稱> #移除套件和設定檔
```


範例：

```
dbtsai@ubuntu:~$ sudo dpkg -P filezilla
正在刪除 filezilla ...
正在清除 filezilla 的設定檔案 ...
dbtsai@ubuntu:~$ dpkg -l | grep filezilla
dbtsai@ubuntu:~$
```

5.1.5 DPKG 與 RPM 共舞


雖然在 Ubuntu 裡面已經提供了很多的套件可以使用，但有一些商業軟體只有提供給 RedHat Linux 或 Fedora 等使用的 RPM 包裝套件。這時候就需要 alien 出馬啦！筆者要提醒讀者，用這一招成功率不是百分之一百喔！

首先要先安裝 alien 這個軟體


 指令：

```
sudo apt-get install alien #安裝 alien
```


簡易的使用方法有下面幾種：

 指令：


```
sudo alien -i <.rpm 套件檔> # 安裝副檔名為 rpm 的套件檔
```

 指令：


```
sudo alien -d <.rpm 套件檔> # 把 rpm 的套件檔轉成 deb 套件檔 (需要 root 身份)
```

 範例：

```
dbtsai@ubuntu:~/deb$ sudo alien -d filezilla-3.0CVS20060622-2.ppc.rpm
filezilla_3.0CVS20060622-3_powerpc.deb generated
```

 指令：


```
alien -r <.deb 套件檔> # 把 deb 的套件檔轉成 rpm 套件檔
```

 範例：


```
dbtsai@ubuntu:~/deb$ alien -r filezilla_3.0CVS20060622_powerpc.deb
filezilla-3.0CVS20060622-2.ppc.rpm generated
```

5.1.6 其他常用的功能


或許有人和筆者一樣有潔癖，想要了解系統上是否有無用的函式庫？主要會發生這種問題是因為當用 apt-get 裝好一套軟體後，它會順便把相依的函式庫裝起來，但是在使用 apt-get 移除那套軟體時，並不會把它相依的函式庫刪除，所以最後系統會累積很多用不到的函式庫！讀者可以透過 deborphan 來察看無用的函式庫。

 安裝指令：

```
sudo apt-get install deborphan # 安裝 deborphan
```

 使用指令：


```
deborphan # 查詢無用的函式庫
# 可以加上 -s 來顯是從哪一個套件庫來的，-P 顯示套件的重要性，-z 顯示套件大小
```

 範例：

```
dbtsai@ubuntu:~$ deborphan -sPz
      48 main/libs          libglut3          optional
     1641 main/libs          wx2.6-ii18n       optional
      712 main/libs          libgnutls11       important
dbtsai@ubuntu:~$
```


它所列出來的套件，基本上是系統已經用不到的，所以可以使用 `dpkg -P` 去把它們移除。

有一些套件在安裝的時候，會自動跑出來對話框讓你設定一些選項。例如 `postfix` 這個郵件伺服器，用 `apt-get install postfix` 安裝後，會讓你在對話框設定一些基本的屬性。若使用者想要重新設定的話，可以透過以下指令：

 指令：

```
sudo dpkg-reconfigure <套件名稱> #重新設定已經安裝的套件
```

而常常會遇到需要重新設定的套件還有 `xserver`，舉例來說讀者可以透過以下指令來重新設定你視窗系統所使用的顯示卡和解析度等選項。

 指令：

```
sudo dpkg-reconfigure xserver-xorg #重新設定 xserver
```


這個功能不是對所有套件都有用的，只對那些有使用到 `debconf` 的套件才能使用喔！

5.2 修改 `/etc/apt/sources.list` 設定檔來增加新的套件源

首先要大家回憶一下，在第一章筆者有提及 Ubuntu 的套件庫為了不同使用者的需求，把套件庫拆成 `main`，`restricted`，`universe` 和 `multiverse` 吧！這些套件

源設定就是負責告訴前端管理系統要從哪裡來抓取套件來源的地方。有些非官方套件也會提供這樣的套件庫來源讓大家安裝套件哩！

這些相關的設定就是在 /etc/apt/sources.list 裡面喔！先用一個你熟悉的文字編輯器把它打開瞧瞧吧！

 指令：

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak # 預防萬一，先備份哩
sudo gedit /etc/apt/sources.list # 開始編輯
```

打開後，可以看到類似下面的設定檔吧！下面那個設定檔為了減少版面，有經過一些精簡，所以有可能和你的有稍微不同喔！如果在一行的最前面是 # 這就代表是註解，也就是管理系統將會忽略這行。

```
deb http://tw.archive.ubuntu.com/ubuntu/ edgy main restricted
deb-src http://tw.archive.ubuntu.com/ubuntu/ edgy main restricted

deb http://tw.archive.ubuntu.com/ubuntu/ edgy-updates main restricted
deb-src http://tw.archive.ubuntu.com/ubuntu/ edgy-updates main restricted

# deb http://tw.archive.ubuntu.com/ubuntu/ edgy universe multiverse
# deb-src http://tw.archive.ubuntu.com/ubuntu/ edgy universe multiverse

# deb http://tw.archive.ubuntu.com/ubuntu/ edgy-backports main restricted
# deb-src http://tw.archive.ubuntu.com/ubuntu/ edgy-backports main restricted

# deb http://tw.archive.ubuntu.com/ubuntu/ edgy-backports universe multiverse
# deb-src http://tw.archive.ubuntu.com/ubuntu/ edgy-backports universe
multiverse

deb http://security.ubuntu.com/ubuntu edgy-security main restricted
deb-src http://security.ubuntu.com/ubuntu edgy-security main restricted
# deb http://security.ubuntu.com/ubuntu edgy-security universe
# deb-src http://security.ubuntu.com/ubuntu edgy-security universe
```

在 Ubuntu 中，預設是把 universe，multiverse 和 backport 關閉的，但是有很多好用的軟體不是在 Ubuntu 預設的 main 和 restricted 之中，例如 gcin 和 pcmanx 等，所以我們先把註解拿掉吧！往後有些套件我們將直接用 apt-get 安裝，不會

再另外說它是屬於哪一個套件源的軟體，如果有遇到不能裝的，記得再來檢查一下該把註解拿掉的地方有沒有拿掉喔！

以下面這個套件源來舉例

```
deb http://tw.archive.ubuntu.com/ubuntu/ edgy main restricted
deb-src http://tw.archive.ubuntu.com/ubuntu/ edgy main restricted
```

表 5.3 套件源設定檔格式

類型	主機位置	版本	包含的套件庫
deb (二進位 套件)	http://tw.archive.ubuntu.com/ubuntu/	edgy	main restricted
deb-src (套 件原始檔)	http://tw.archive.ubuntu.com/ubuntu/	edgy	main restricted

如表 5.3 所示，這兩個套件源可以歸納成這四個欄位，主機位置那個欄可以設置成離你位置最近的地方，例如 Ubuntu Taiwan 提供的 <http://atp.ubuntu.org.tw/ubuntu/> 鏡像站。而 deb-src 的那一行，是提供套件的原始碼清單，若你不需要用 apt-get 系統去抓原始檔，你可以把它註解掉。你可以在有新版本的時候，把版本欄位上的 edgy 改成下一版的研發代號，那麼再執行以下指令，那麼就會變成最新版喔！

指令：

```
sudo apt-get update          # 更新套件庫清單
sudo apt-get dist-upgrade -y # 開始上網抓新套件升級
```

而版本的地方額外有 edgy-updates，edgy-backports 和 edgy-security，這三者的意義筆者在第一章介紹 Ubuntu 的地方有說明過了，如果讀者忘了，可以去翻一下。


由上面的規則你也可以了解到這兩行套件庫設定

```
deb http://tw.archive.ubuntu.com/ubuntu/ edgy main restricted
deb http://tw.archive.ubuntu.com/ubuntu/ edgy universe multiverse
```

其實可以整合成以下一行。

```
deb http://tw.archive.ubuntu.com/ubuntu/ edgy main restricted universe
multiverse
```

提醒你，每次改過 `sources.list`，都要再執行一次以下指令來更新清單喔！

 指令：

```
sudo apt-get update # 更新套件庫清單
```

建議讀者，不要在 `sources.list` 裡面混用不同版本的二進位套件庫。例如混用 Debian 的套件庫或 Ubuntu 新舊版混用。因為新舊版套件可能會因為相依性不同而讓一堆軟體不能使用，甚至把系統搞掛。若是你像筆者一樣，喜歡用最新的軟體，但又不想把 Ubuntu 升級到研發的測試版，那可以考慮和筆者一樣，把 `deb-src` 那一行的版本改成測試版，然後自己從原始碼來編譯。這樣可以控制只單純升級某一套軟體，不會有混用版本所衍生的問題。


5.3 APT (Advanced Package Tool)

談到 Debian like 的發行版，最為人稱道的就是 APT (Advanced Package Tool) 套件管理系統。它可以很簡單的讓使用者線上安裝軟體並且自動解決相依性問題，也可以幫助使用者升級系統，以及使用關鍵字搜尋未安裝的套件等功能。它底層是以 DPKG 來達成，但它能幫助我們處理很多直接使用 `dpkg` 的煩瑣細節，所以這個工具讀者要好好研究。


5.3.1 apt-cache 查詢可用套件

讀者可以把這一系列指令想成是類似 DPKG 查詢的功能，不過它更為強大，可以針對還沒有安裝的軟體作查詢。

這是最基本也是最常用的指令，我們可以透過它以關鍵字來尋找想要安裝的軟體：

 指令：

```
apt-cache search <關鍵字> # 從套件資訊中查詢包含關鍵字的套件
```

 範例：

```


dbtsai@ubuntu:~$ apt-cache search latex
autotex - An integrated environment for writing TeX/LaTeX documents
debiandoc-sgml - DebianDoc SGML DTD and formatting tools
doxygen - Documentation system for C, C++, Java, Python and other languages
doxygen-doc - Documentation for doxygen
hevea - translates from LaTeX to HTML, info, or text
ispell - International Ispell (an interactive spelling corrector)
以下略
dbtsai@ubuntu:~$

```

由於這個方式所做的搜尋是從套件資訊裡面尋找，而不是只從套件名稱，所以往往會找到許多不是讀者所期望的資訊。故我們可以如以下範例，善用 `grep` 來作一些條件搜尋喔！

舉例來說，上一個例子搜尋到幾十個和 `latex` 相關軟體，如果我們是想要找 `latex` 字型，那就可以如以下範例加上條件。

備註 `apt-cache` 預設是忽略大小寫的，但 `grep` 是有分大小寫，如果你要 `grep` 也不分大小寫，可以加上 `-i` 的參數喔！


 範例：

```


dbtsai@ubuntu:~$ apt-cache search latex | grep font
latex-xft-fonts - Xft-compatible versions of some LaTeX fonts
tfm-arphic-bkai00mp - Arphic "AR PL KaitiM Big5" TrueType font TeX font metric data
tfm-arphic-bsmi00lp - Arphic "AR PL Mingti2L Big5" TrueType font TeX font metric data
tfm-arphic-gbsn00lp - Arphic "AR PL SungtiL GB" TrueType font TeX font metric data
tfm-arphic-gkai00mp - Arphic "AR PL KaitiM GB" TrueType font TeX font metric data
ttf2pt1-chinese - Chinese fonts encoding maps for ttf2pt1
dbtsai@ubuntu:~$

```

若是想單純的只在套件名稱內以關鍵字搜尋，那麼可以試試看這個方法。此時搜尋到的套件會比較少，也就是所得到的套件名稱都有包含關鍵字喔！


 指令：

```
apt-cache -n search <關鍵字> # 從套件名稱搜尋關鍵字
```


 範例：

```
dbtsai@ubuntu:~$ apt-cache -n search latex
latex-ucs - support for using UTF-8 input encoding in LaTeX documents
latex-ucs-contrib - additional languages for latex-ucs
latex-ucs-dev - configuration source files for latex-ucs
latex-ucs-doc - documentation for latex-ucs
latex-ucs-uninames - full unicode character names for latex-ucs
以下略
dbtsai@ubuntu:~$
```

如果你找到想要安裝的軟體，但是想要先了解某個套件詳細的資訊和介紹，可以透過以下指令來得到詳細的資訊：

 指令：

```
apt-cache show <套件名稱> # 顯示套件詳細的資訊和介紹
```

 範例：

```
dbtsai@ubuntu:~$ apt-cache show wxvlc
Package: wxvlc
Priority: optional
Section: universe/graphics
Installed-Size: 1208
Maintainer: Sam Hocevar (Debian packages) <sam+deb@zoy.org>
Architecture: powerpc
Source: vlc
Version: 0.8.4.debian-1ubuntu6
Depends: vlc (= 0.8.4.debian-1ubuntu6), libc6 (>= 2.3.4-1), libcdio6, libgcc1 (>= 1:4.0.2), libiso9660-4, libstdc++6 (>= 4.0.2-4), libvcdinfo0 (>> 0.7.23)
Filename: pool/universe/v/vlc/wxvlc_0.8.4.debian-1ubuntu6_powerpc.deb
Size: 357804
MD5sum: 527a50e2f4f18d3109c831d2e6293c71
Description: wxWidgets frontend for VLC
```

VLC is the VideoLAN project's media player. It plays MPEG, MPEG2, MPEG4,

```
DivX, MOV, WMV, QuickTime, mp3, Ogg/Vorbis files, DVDs, VCDs, and multimedia streams from various network sources.
```

```
dbtsai@ubuntu:~$
```

記不記得我們前面談過 `sources.list` 有分 `deb` 和 `deb-src` 兩種套件源呢？其中第二個就是包含原始碼資訊的套件源。讀者可以透過以下指令來取得原始碼資訊，例如有編譯相依套件等訊息。常常套件原始碼只有一個，但是為了讓使用可以選擇需要的部份，編譯後會產生數個 `deb` 可安裝二進位套件，其中 `binary` 那一欄就是紀錄此資訊的地方：


指令：

```
apt-cache showsrc <套件名稱> # 顯示套件原始碼相關資訊
```


範例：

```
dbtsai@ubuntu:~$ apt-cache showsrc wxvlc
Package: vlc
Binary: vlc-plugin-glide, vlc, vlc-plugin-alsa, vlc-plugin-svgalib,
vlc-plugin-ggi, vlc-plugin-esd, vlc-plugin-arts, vlc-plugin-sdl, wxvlc,
libvlc0-dev, mozilla-plugin-vlc
Version: 0.8.5.debian-2ubuntu1
Priority: optional
Section: universe/graphics
Build-Depends: debhelper (>= 4.0), dh-buildinfo, quilt, nasm, yasm [amd64],
gettext, linux-kernel-headers (>= 2.5.999-test7-bk-7), firefox-dev,
libgl1-mesa-dev
略
Architecture: any
Standards-Version: 3.7.2
Directory: pool/universe/v/vlc
Files:
ebfb04e8ca231a27e42c48b6d7c55c1d 1923 vlc_0.8.5.debian-2ubuntu1.dsc
9cb8acc0341e59b05bb00d10a504f0d1 14045323 vlc_0.8.5.debian.orig.tar.gz
f6edb80536f618f5eee4c9667bd02829 9809 vlc_0.8.5.debian-2ubuntu1.diff.gz
```

若要查詢軟體安裝的相依性等資訊，可以透過以下指令查詢。其中 `Reverse Depends` 代表了反向相依性，它列出的軟體都是安裝時都會先裝被查詢的套件。`Dependencies` 代表著安裝被查詢套件前，所需要先安裝的套件。


 指令：

```
apt-cache showpkg <套件名稱> # 顯示套件相依性等相關資訊
```


 範例：

```
dbtsai@ubuntu:~$ apt-cache showpkg wxvlc
Package: wxvlc
Reverse Depends:      # 裝這些軟體都會先裝 wxvlc 喔！
    vlc,wxvlc
    gvlc,wxvlc
    gnome-vlc,wxvlc
Dependencies:         # 安裝 wxvlc 以前，都要先裝以下軟體呢！
0.8.4.debian-lubuntu6 - vlc (5 0.8.4.debian-lubuntu6) libc6 (2 2.3.4-1) libcdio6
(0 (null)) libgcc1 (2 1:4.0.2) libiso9660-4 (0 (null)) libstdc++6 (2 4.0.2-4)
libvcdinfo0 (4 0.7.23) libwxgtk2.6-0 (2 2.6.1.2ubuntu2)
dbtsai@ubuntu:~$
```

若覺得用以上方式列印出來的相依性資訊有點亂，可以試試看用這個指令。這個指令只會列出此軟體相依的套件，並且用比較簡單明瞭的方式列印出來：


 指令：

```
apt-cache depends <套件名稱> # 顯示套件相依性等相關資訊
```


 範例：

```
dbtsai@ubuntu:~$ apt-cache depends wxvlc
wxvlc
  依存關係: vlc
  依存關係: libc6
  依存關係: libcdio6
  依存關係: libgcc1
  依存關係: libiso9660-4
  依存關係: libstdc++6
  依存關係: libvcdinfo0
  依存關係: libwxgtk2.6-0
dbtsai@ubuntu:~$
```

上一個範例沒有列出反向相依的軟體，除了用 `apt-cache showpkg` 來同時列出相依和反相依套件外，可以用此指令列出反相依套件：


 指令：

```
apt-cache rdepends <套件名稱> # 顯示套件反向相依性等相關資訊
```

 範例：

```
dbtsai@ubuntu:~$ apt-cache rdepends wxvlc
wxvlc
Reverse Depends:
  vlc
  qvlc
  kvlc
  gvlc
  gnome-vlc
dbtsai@ubuntu:~$
```

可以透過以下指令來了解系統套件庫上總共有多少可用的套件：

 範例：

```
dbtsai@ubuntu:~$ apt-cache stats
所有套件的名稱 23606 (944k)
  普通套件：18259
  虛擬套件：338
  單虛擬套件：1043
  混合虛擬套件：187
  失蹤：3779
所有不同版本 19218 (922k)
所有依存關係：132990 (3724k)
所有版本/檔案關聯：20688 (331k)
所有提供套件對應：3153 (63.1k)
Glob 字串共計：119 (1521)
所有相依版本空間：576k
Slack 空間共計：113k
所有統計後的空間：6099k
dbtsai@ubuntu:~$
```

5.3.2 apt-get 套件升級，安裝和移除

這系列的指令，大概會是你使用 Ubuntu 最常用的吧！例如筆者每天有空沒空都會跑一下以下範例來自動更新，若有安全性的問題 Ubuntu 也會在第一時間幫你解決呢！所以這麼好用的東西，我們來好好瞧瞧吧！

範例：

```
dbtsai@ubuntu:~$ sudo apt-get update
下載:1 http://security.ubuntu.com dapper-security Release.gpg [189B]
下載:2 http://apt.ubuntu.org.tw dapper Release.gpg [189B]
略
dbtsai@ubuntu:~$ sudo apt-get upgrade
了解套件依存關係中... 完成
下列的套件都將更新:
  gedit gedit-common gnome-about gnome-desktop-data gnome-games gnome-games data
  gnome-panel gnome-panel-data
更新 18 個套件，新安裝 0 個套件，刪除 0 個套件，另不更新 0 個套件。
略
dbtsai@ubuntu:~$
```

首先，這個指令已經在前面出現了好幾次了，讀者大概已經知道在幹嘛了吧！不過在這裡正式的介紹它一下！當我們更新 `sources.list` 後，就要用這個指令同步系統上和伺服器上的套件清單（package list）。或者當一陣子沒有做這個動作時，也要更新一下，因為伺服器上的清單可能已經改變了。假設你沒有做這個動作，而恰巧伺服器上有新的安全更新，那就算你 `apt-get upgrade` 也不會更新到那些新版的套件喔！筆者個人的習慣是每次 `apt-get upgrade` 以前就會先跑一次 `apt-get update`，反正也不用多少時間。

指令：

```
apt-get update # 更新系統套件庫清單
```

這個指令會依照 `/etc/sources.list` 裡面所列的套件源去抓套件庫，然後會把相關資訊存在 `/var/lib/apt/lists/*` 裡面，讀者可以透過文字編輯器看到目錄裡下各檔案裡面的資訊！

升級有兩種，第一種是比較保守安全的升級。怎們說呢？透過以下指令，它在升級的時候，不會因為要滿足相依性而動到其他套件。例如現在 A 軟體是 1.0 版，A 相依的套件是 B 和 C。現在系統上裝了 A，所以當然因為相依性而裝了 B 和 C。那天 A 出了新版 2.0，它相依的套件是 C 和 D，且 D 和 B 衝突。那麼若要更新的話，勢必得要安裝 D 且移除 B。這種更新方式是不會幫你自動做到這些的，它會幫你自動處理那些不會在更新過程中動到其他套件的升級。因此它是相對安全的更新，但使用者不用擔心，若是在同一個版本內更新，例如 sources.list 都是 edgy 時做安全等更新，基本上系統變動不會太大，所以用這種更新就夠了喔！

指令：


```
sudo apt-get upgrade          # 更新沒有相依性問題的系統套件，通常用在常規更新
# 若有升級檔時，系統會問你是否確定升級。但若寫在 crontab 自動更新時，
# 這樣會很不方便，所以讀者可以加上 -y 的參數來自動回答 yes。
# -s 模擬升級，此時會測試升級過程，但並不會實際升級你的系統。
```

若是從 6.06 (dapper) 升級到 6.10 (edgy)，系統的變動幅度可能很大，此時有些新版的套件可能需要有額外的相依套件，或者因為虛擬套件 (metapackage) 相依的套件名稱改變等，此時系統升級的時候需要處理新的套件安裝和舊的套件移除，那個就得要用這種升級方式。由於可能會安裝或移除其他套件，因此這個方法有一點風險。

指令：

```
sudo apt-get dist-upgrade    # 更新所有系統套件，若有相依性問題則解決，
# 若有升級檔時，系統會問你是否確定升級。但若寫在 crontab 自動更新時，
# 這樣會很不方便，所以讀者可以加上 -y 的參數來自動回答 yes。
# -s 模擬升級，此時會測試升級過程，但並不會實際升級你的系統。
```

當升級到測試版時，或者安裝新套件後，也可能在多種版本的二進位檔混用時，會造成系統錯誤（通常只有在安裝測試版的 Ubuntu 才會發生）。這個時候會產生很多錯誤訊息，然後整個套件管理系統開始抓狂，此時讀者大概也會想要敲壞電腦！其實這個時候可以死馬當活馬醫，當把套件管理系統搞掛了以後，可以試試看以下四個指令重複搭配使用，有時候真的能救回來喔！那個 -f 所代表的意義就是遇到錯誤或嚴重的相依性問題也去試圖修正，這個參數在 apt-get install 或 apt-get remove 也可以使用。


 指令：

```

sudo apt-get update           # 更新套件清單
sudo apt-get -f -y upgrade    # 遇到錯誤也繼續 upgrade
sudo apt-get -f -y dist-upgrade # 遇到錯誤也繼續 dist-upgrade
sudo dpkg --configure -a     # 重新設定之前因中斷等還沒安裝好的套件

```


當讀者要安裝軟體時，確定了套件名稱以後，可以使用以下指令。它還可以順便幫你安裝所相依的相關套件呢！

 指令：

```

sudo apt-get install <套件名稱> # 自動安裝所指定的套件
# 讀者可以透過 -y 跳過確定詢問。-f 強迫遇到錯誤也試著修正安裝。
# -s 模擬安裝，此時會測試安裝過程，但並不會實際安裝到你系統上。
# 有時候某套件安裝錯誤，但是它的重要性很高，我們無法把它刪除後
# 再重新安裝，那我們可以透過 --reinstall 的參數來達成。而 -d 的參數是
# 只會下載 deb 套件，套件檔案將會放在 /var/cache/apt/archives 裡面。

```

 範例：

```

dbtsai@ubuntu:~$ sudo apt-get install wxvlc
下列的【新】套件都將被安裝：
  vlc vlc-plugin-alsa
建議 (Suggested) 的套件：
  ttf-thryomanes mozilla-plugin-vlc
推薦 (Recommended) 的套件：
  videolan-doc
下列的【新】套件都將被安裝：
  vlc vlc-plugin-alsa wxvlc
更新 0 個套件，新安裝 3 個套件，刪除 0 個套件，另不更新 0 個套件。
略
dbtsai@ubuntu:~$

```

在 5.1.4 節提到，用 dpkg 移除軟體時，有一種是只移除套件但不移除設定檔，此時套件的狀態是 rc。另外一種是完全移除，也就是套件和設定檔都將會被移除！所以用 APT 來移除套件也是有分這兩種。而用 apt-get remove 比用 dpkg -r 或 dpkg -R 還要方便的地方是，當套件 A 相依於套件 B，套件 B 相依於套件 C，若我們要移除套件 C，使用 dpkg 系統移除套件 C 時，會出現警告，告訴你因為相依性錯誤不讓你移除。此時你得要從 A 套件，B 套件然後是 C 套件這樣的順序

慢慢的用 `dpkg` 去移除。若是採用 `apt` 系統來移除套件，那麼可以直接移除掉套件 C，且系統會自動計算相依性，也把套件 A 和 B 移除喔！

指令：

```
sudo apt-get remove <套件名稱> # 自動移除所指定的套件，但留下設定檔
# 讀者可以透過 -y 跳過確定詢問，-f 強迫遇到錯誤也試著修正移除。
# -s 模擬移除，此時會測試移除過程，但並不會實際移除套件。
```

指令：


```
sudo apt-get --purge remove <套件名稱> # 自動移除所指定的套件，且移除設定檔
# 讀者可以透過 -y 跳過確定詢問，-f 強迫遇到錯誤也試著修正移除。
# -s 模擬移除，此時會測試移除過程，但並不會實際移除套件。
```

範例：


```
dbtsai@ubuntu:~$ sudo dpkg -r wxvlc
dpkg：由於相依問題的緣故，wxvlc 不能被刪除：
    vlc 相依於 wxvlc。
dpkg：處理 wxvlc (--remove)時出錯：
    相依性出錯 - 將不刪除
在處理時有錯誤發生：
    wxvlc
dbtsai@ubuntu:~$ sudo apt-get remove wxvlc
了解套件依存關係中... 完成
下列的套件都將被【刪除】：
    vlc vlc-plugin-alsa wxvlc
正在刪除 wxvlc ...
正在刪除 vlc ...
正在刪除 vlc-plugin-alsa ...
dbtsai@ubuntu:~$
```

當安裝套件和系統更新時，這些套件會先被下載到 `/var/cache/apt/archives` 的快取目錄，然後再被安裝。在安裝後這些暫存的套件不會自動刪除。這個立意本來是良好了，是為了某些頻寬不足的用戶可以直接拷貝這個目錄底下的套件檔到其他機器，而不需要重新下載。但若你覺得這些檔案太占空間，不需要再保留，可以考慮把它們移除。移除的方式有兩種，第一種是完全移除，第二種是保留當


前系統上用到的套件，而只刪除舊版的。筆者個人是想到就會去 `apt-get clean` 一下！

 指令：

```
sudo apt-get clean # 移除所有的套件暫存檔
```

 指令：

```
sudo apt-get autoclean # 移除舊版的所有套件暫存檔
```

 範例：

```
dbtsai@www:~$ ls /var/cache/apt/archives/
apache2-common_2.0.55-4ubuntu2.1_i386.deb
apache2-mpm-prefork_2.0.55-4ubuntu2.1_i386.deb
firefox_1.5.dfsg+1.5.0.5-0ubuntu6.06_i386.deb
略
dbtsai@www:~$ sudo apt-get clean
dbtsai@www:~$ ls /var/cache/apt/archives/
lock partial
dbtsai@www:~$
```

基本上了解以上內容，常用的 `apt-get` 功能應該沒有問題了。它還有一些關於下載原始碼和從原始碼直接編譯成套件的功能，因為是進階且初學者不常用到，和這邊也無法成為很連貫的章節，所以筆者將這些材料獨立到 5.5 節。

5.3.3 apt-file 以檔案找套件

先前提過，可以透過 `dpkg -S` 來用系統上的檔案來查詢是哪個套件提供的。但這個方式是針對已經安裝的套件才有效。遇到以下慘劇時，通常是缺少還沒安裝的套件所提供的檔案，此時 `dpkg -S` 就行不通了。假如我們在編譯程式時，系統和我們抱怨缺一堆 `.h` 的表頭檔，除了用 `google` 找是那個套件提供的，還有什麼辦法嗎？換個場景，熟悉網管的人，一定常常寫 `shell script` 來管理自己的主機，有時候會希望把一些紀錄檔 E-Mail 到自己的帳號，以便掌握系統狀態。

通常在 `shell script` 裡面我們都配合 `mail` 這個指令來寄信，但是讀者會發現在 Ubuntu 內，預設根本沒有這個指令啊！用 `apt-cache search mail`，出現了一大

堆套件，天啊！誰知道要裝那個啊！這時候這個功能就很有用，我們可以透過已知的執行檔、文件、函式庫、標頭檔等用檔案來找套件喔！

首先要安裝 apt-file 這套程式。

指令：

```
sudo apt-get install apt-file
```

類似於 apt-get，它在初次使用的時候也要和伺服器同步所有套件的資訊清單，而它根據的伺服器清單也是 sources.list 喔！

指令：

```
sudo apt-file update
```

這支程式運作的方式是把所有套件的檔案列表抓出來，然後比對的時候是以檔案路徑加上檔名做為被關鍵字比對的字串。

指令：

```
apt-file search <關鍵字>          # 以關鍵字搜尋套件
# 讀者可以加上 -l 來只顯示套件，而不顯示檔名，不過既然是以檔名找套件，
# 當然是顯示檔名比較方便啦！因為是對檔案名稱比對，所以預設是有區分大小寫，
# 讀者可以加上 -i 的參數，來不分大小寫喔！而 -F 可以讓搜尋的時候讓關鍵字
# 和被比對對象完全吻合。
```

首先試試看比較容易搜尋到正確套件的關鍵字，wx.h。這是開發跨平台視窗程式所需要的一個表頭檔。依照底下的範例，得知我們需要裝的是 wx2.6-headers。

範例：

```
dbtsai@ubuntu:~$ apt-file search wx.h
wx2.4-examples: usr/share/doc/wx2.4-examples/examples/samples/help/doc/wx.htm
wx2.4-headers:  usr/include/wx/wx.h
wx2.6-examples: usr/share/doc/wx2.6-examples/examples/samples/help/doc/wx.htm
wx2.6-headers:  usr/include/wx-2.6/wx/wx.h
dbtsai@ubuntu:~$
```


但若是像本節開頭那個例子，我們要找 mail 這支程式在哪個套件，用一樣的方式搜尋，就似乎沒有那麼好運了！

範例：

```
dbtsai@ubuntu:~$ apt-file search mail
a2ps: usr/share/a2ps/sheets/mail.ssh
abiword-help:
usr/share/AbiSuite-2.4/AbiWord/help/en-US/howto/howtomailmerge.html
abiword-help:
usr/share/AbiSuite-2.4/AbiWord/help/en-US/interface/dialogmailmerge.html
略過上百行！！
dbtsai@ubuntu:~$
```

你會發現結果有上百個阿！只要路徑或檔名有 mail 這個關鍵字的，都會被搜尋出來，除非你很有耐心的一個一個找，不然這種資訊是沒用的。絕大部份的執行檔都是放在 /usr/bin 裡面，所以我們可以試著猜檔案的絕對位置是 /usr/bin/mail，這次試試看用它來當關鍵字。

範例：

```
dbtsai@ubuntu:~$ apt-file search /usr/bin/mail
courier-base: usr/bin/maildiracl
courier-base: usr/bin/maildirkw
esmtplib-run: usr/bin/mailq
exim4-daemon-heavy: usr/bin/mailq
以下略
maildir-filter: usr/bin/maildir-filter
```


這次少了很多了，只有數十個，但找到的都是關鍵字後面可以加上任意字串，所以還是不是我們想要的東西。現在可以試著加上 -F 的參數，讓字串完全正確的比對，這樣答案就出來了唷！

範例：


```
dbtsai@ubuntu:~$ apt-file search /usr/bin/mail -F
mailutils: usr/bin/mail
mailx: usr/bin/mail
dbtsai@ubuntu:~$
```

這次就很正確的只有找到兩個，讀者可以分別用 `apt-cache show` 去看看它們的說明，然後決定要裝哪一個。在筆者的伺服器上，我是裝 `mailx`，這樣就能透過指令來寄系統紀錄檔到筆者的信箱，尤其對筆者這種管了很多台伺服器的人特別好用喔！

最後這個功能，其實就是類似 `dpkg -L`，不過它能列出還未安裝的套件所包含的檔案。

 指令：

```
sudo apt-file list <套件名稱> # 列出套件所包含的檔案
```

 範例：

```
dbtsai@ubuntu:~$ apt-file list wxvlc
wxvlc: usr/bin/wxvlc
wxvlc: usr/lib/vlc/gui/libwxwidgets_plugin.so
wxvlc: usr/share/applications/vlc.desktop
wxvlc: usr/share/doc/wxvlc
略
dbtsai@ubuntu:~$
```

5.3.4 apt-key 金鑰管理

若讀者不在 Ubuntu 裡面使用額外的非官方套件源，或對於套件安全性沒興趣者，這一小節可以略過，不會減少對於後面學習的連貫性喔，或者有空在回來慢慢品嚐吧！

讀者或許會好奇，Ubuntu 怎麼能保證全世界那麼多台 mirror 站提供的套件不是經過有心的駭客入侵更改，或者駭客自己架一台 mirror 站，再用 DNS 挾持的方式，把你引導到他自己架的套件庫上，並在你系統更新時順便植入後門。

在 Debian 和 Ubuntu 採用簽章的方式來解決這個問題。簽章可以證明這些套件檔是由 Ubuntu 官方所出，而沒有經過別人所篡改，並且這個簽章是跟著套件一起跑，不管跑到哪一個 mirror 站，都無法改變的。如果這個套件檔被有心人士修改加入後門，那麼這個簽章將會失效，此時 APT 系統就會判別出這個套件經過人家改動，而提出警告。

所謂的簽章，是透過金鑰的功能來達成。完整的金鑰有兩隻鑰匙，一隻是私鑰（Private Key），另外一隻是公鑰（Public key），而公鑰和私鑰是一對的，並且是一起產生的。金鑰加密的基礎是，當我們用私鑰對一段訊息加密後，我們只能用公鑰把它解開，反之如此。因此其實公鑰和私鑰其實地位是一樣的，就看你要把哪一把當成公鑰，哪一把當成私鑰。這種加密方式就是所謂的非對稱式密碼系統。當然讀者或許會覺得奇怪，理論上我們知道一把公鑰後，應該是可以算出一把私鑰和這把公鑰吻合，這樣不就破解了嘛？的確如此，不過以一把鑰匙算出成對的另外一把鑰匙是幾乎不可能在有限的時間內計算出來的，以現在的電腦能力，破解一把高位元的鑰匙，都要數十年的時間，所以這種加密的安全性主要是依賴於從公鑰推出私鑰的計算所需的時間。但沒有人保證明天的電腦科技不會突飛猛進，造成可以在合理的時間內破解金鑰，但目前這種加密方式仍然是很可靠的。

我們如何透過這種系統來進行秘密通訊呢？舉例 A 和 B 要進行秘密通訊，而 A 和 B 的公鑰都已經公佈出來，彼此都知道了。此時 A 可以用 B 的公鑰把要傳給 B 的訊息加密，如此這段訊息只有用 B 的私鑰才能解開，因而達到秘密通訊的要求。同理，B 要傳秘密訊息給 A，就用 A 的公鑰加密，如此就只有 A 能解開這段文字。這也是為什麼私鑰很重要了，這種加密方式安全度就看私鑰有沒有保管好。總之，公鑰可以公佈出去，給越多人知道也沒關係，但私鑰就要自己好好保管，絕不能外洩。

在套件管理系統中，採用的簽章系統加密的想法和前面秘密通訊剛好反向。若要確定檔案是從官方來的，此時官方組織會用私鑰加密，這時只有公鑰能夠解密。當 Ubuntu 用私鑰加密一段文字後，使用者可以用 Ubuntu 的公鑰去把它解開來。但若是駭客沒有 Ubuntu 的私鑰，隨使用個鑰匙加密，那麼我們用 Ubuntu 的公鑰去解密，會出現失敗。因此只要我們的公鑰是正確的，我們就可以判斷檔案或文字是否是由 Ubuntu 官方出來的。

問題是透過私鑰加密的檔案會變大，且隨著檔案大小越大，用公鑰解密的時間越來越長，所以把所有套件檔加密是不切實際的。這裡有個變通的方式，是讓所有的人不管有沒有公鑰，都可以知道檔案內容，也就是現在套件檔不加密，只要有辦法確定是官方的就好了。這就是所謂的數位簽章系統，可以保證資料是你所撰寫，類似於我們簽名，雖然內容大家都看的到，但是可以用筆跡來認出是誰簽名的。方法是我們對於要簽章的檔案作單向 hash，例如用 MD5 hash，然後把

hash 後的值用私鑰加密。而驗證的人只要用公鑰確定那段被加密的 MD5 hash 是否是由私鑰產生的，並且可以把它解密後再對文件做一次 MD5 hash，如果兩個 hash 的值都一樣，那代表是本人寫的，沒有被篡改過。

實際上 Ubuntu 的套件管理系統是用 MD5 和簽章系統來配合做驗證。所有的套件檔都會產生 MD5 hash，這些 MD5 碼可以檢查套件是否完整，是否被改過。當套件被修改過而產生一樣的 MD5 碼是幾乎不可能的，所以我們可以用此值來驗證套件的完整性。現在的問題是如何安全傳遞這些 MD5 hash 值，方法就是簽章系統了。在 APT 系統中，當我們做 `apt-get update` 時，會向套件庫取得兩個檔案，`Release` 和 `Release.gpg`。在 `Release` 裡面包含了套件庫所有 deb 套件的 MD5 hash，而保障 `Release` 是沒被篡改過的方法是透過 `Release.gpg`，也就是簽章。`Release.gpg` 包含了官方對 `Release` 做單向 hash 後再用私鑰加密的資訊。因此我們可以用公鑰來驗證 `Release.gpg` 是否由私鑰所加密的，並用解出來的資訊再對 `Release` 驗證是否被篡改過。

這些理論在資工系密碼學都要學一學期，所以這裡只能簡略的帶過。重點是只要知道它能保障你下載的套件檔是沒有被篡改過的就行了。以下我們要說明實務上要如何管理 APT 系統的公鑰。

我們先列出 APT 系統目前有的公鑰看看，當然 Ubuntu 套件庫所用的公鑰在你安裝好系統後，就已經裝好啦！

指令：

```
sudo apt-key list # 列出 APT 系統所使用的公鑰
```

範例：

```
dbtsai@ubuntu:~$ sudo apt-key list
/etc/apt/trusted.gpg
-----
pub 1024D/437D05B5 2004-09-12 # 1024D 此欄代表著金鑰擁有者 ID，437D05B5 此欄是金鑰指紋
uid Ubuntu Archive Automatic Signing Key <ftpmaster@ubuntu.com>
sub 2048g/79164387 2004-09-12

pub 1024D/FBB75451 2004-12-30
uid Ubuntu CD Image Automatic Signing Key <cdimage@ubuntu.com>
```

```
dbtsai@ubuntu:~$
```

你會發現 APT 系統上目前有兩把公鑰，一把是給光碟用的，另外一把是給套件庫所使用的。而我們系統用得好好的，幹嘛還要裝新的公鑰呢？常常金鑰都會設幾年的保存期限，避免有心人用超級電腦從公鑰算出私鑰。這種金鑰過期的情形在 Debian 就發生過，幾年前 Debian 就換過金鑰導致很多人套件管理系統無法更新。或者我們常常加入個人的套件庫，他們當然沒有 Ubuntu 的私鑰（這可是 Ubuntu 最保密的事情呢！），所以系統上的公鑰就無法驗證這些套件庫，所以我們就得要手動把相對應的公鑰裝上去。

例如筆者很喜歡在我的 Ubuntu 上加上 Debian unstable 和 experiment 的原始碼套件源，這樣我可以透過套件管理系統來自動編譯最新的套件。當然在 Ubuntu 裡面沒有 Debian 的公鑰，如以下範例，會產生錯誤訊息。

範例：

```
dbtsai@ubuntu:~$ cat /etc/apt/sources.list
deb-src http://ftp.debian.org.tw/debian experimental main
deb-src http://ftp.debian.org.tw/debian unstable main

deb http://apt.ubuntu.org.tw/ubuntu/ dapper main restricted
deb-src http://apt.ubuntu.org.tw/ubuntu/ dapper main multiverse

dbtsai@ubuntu:~$ dbtsai@ubuntu:~$ sudo apt-get update
下載:2 http://ftp.debian.org.tw unstable Release.gpg [189B]
下載:9 http://ftp.debian.org.tw unstable Release [38.3kB]
中間略

W: GPG error: http://ftp.debian.org.tw unstable Release: The following signatures
couldn't be verified because the public key is not available: NO_PUBKEY
010908312D230C5F

W: 用「apt-get -f install」指令或許能修正這些問題。
dbtsai@ubuntu:~$
```

此時系統抱怨找不到公鑰 (NO_PUBKEY 010908312D230C5F)。或許讀者會覺得奇怪，公鑰那麼短不是一下子就會被破解了嗎？其實那不是公鑰，而是金鑰

指紋的最後 16 碼。一般高位元的金鑰，公鑰大小會有 2K 以上，這樣若要給別人公鑰除非是用磁片傳，不然無法很容易傳遞。想想看 2K 可以紀錄一千多個中文字耶！總不能唸給別人聽吧！所以有人就想到一個交換公鑰的方式，利用 key server 來傳遞。我們先對金鑰取特徵值，得到一組金鑰指紋，有 40 碼，每一碼都是從 0 到 F，以十六進位表示。然後我們可以把公鑰上傳到 key server，其他人只要憑金鑰指紋就可以下載，這樣就解決了煩人的公鑰交換問題。

例如筆者的金鑰是 DSA + Elgamal 4096bits，而我的鑰匙資訊如下：

筆者的金鑰：

```
pub 1024D/91AFDF6C 2006-08-02
    金鑰指紋 = A612 E53D DE36 5627 4F60 665D 2451 ACB6 91AF DF6C
uid                               Dung-Bang Tsai (dbtsai) <dbtsai@gmail.com>
sub 4096g/A1F68A40 2006-08-02
```

筆者產生了金鑰後，其中的資訊 1024D/91AFDF6C 是我們最常用的。前面的 1024D 是從筆者的名子和 E-Mail 算出來，也就是 User ID。後面的 91AFDF6C 是金鑰指紋後八碼。整個組合起來就是你的金鑰 ID。

如果公鑰已經上傳到 key server 上，我們可以透過金鑰指紋後八碼下載到該金鑰的公鑰。例如讀者可以透過以下指令來下載筆者的公鑰：

指令：

```
gpg --keyserver hkp://wwwkeys.eu.pgp.net --recv-keys <金鑰指紋>
# 伺服器可以選擇其他的，這些伺服器都會互相同步。
# 所以 eu 可以換成 us cz de 等。
# 而金鑰指紋最少要 8 碼，或 16 碼，也可以完整的用 40 碼來下載公鑰。
```

範例：

```
dbtsai@ubuntu:~$ gpg --keyserver hkp://wwwkeys.eu.pgp.net --recv-keys 91AFDF6C
gpg: 正在請求金鑰 91AFDF6C 自 hkp 伺服器 wwwkeys.eu.pgp.net
gpg: 金鑰 91AFDF6C: 公鑰 "Dung-Bang Tsai (dbtsai) <dbtsai@gmail.com>" 已被匯入 gpg:
總共被處理的數量: 1
gpg:                               已匯入: 1
dbtsai@ubuntu:~$ gpg --list-key
/home/dbtsai/.gnupg/pubring.gpg
```

```
-----
pub 1024D/91AFDF6C 2006-08-02
uid          Dung-Bang Tsai (dbtsai) <dbtsai@gmail.com>
sub 4096g/A1F68A40 2006-08-02
dbtsai@ubuntu:~$
```

回到主題，我們先前談及要下載 Debian 套件庫的公鑰，根據錯誤訊息 NO_PUBKEY 010908312D230C5F，後面那一串就是金鑰指紋。我們可以貼上最後 8 碼，或者 16 碼來下載。

備註 通常有在使用金鑰的人都會上傳他的公鑰到伺服器上，若真的沒有上傳的話，就必須和該套件庫的維護者索取。

取得 Debian 套件庫公鑰：

```
dbtsai@ubuntu:~$ gpg --keyserver hkp://wwwkeys.eu.gpg.net --recv-keys
010908312D230C5F
dbtsai@ubuntu:~$ dbtsai@ubuntu:~$ gpg --list-key
/home/dbtsai/.gnupg/pubring.gpg
-----
pub 1024D/2D230C5F 2006-01-03 [過期: 2007-02-07]
uid          Debian Archive Automatic Signing Key (2006) <ftpmaster@debian.org>
dbtsai@ubuntu:~$
```

由於 APT 系統上所使用的公鑰列表和 gpg 系統的公鑰列表是分別獨立的，所以現在要把 gpg 的公鑰匯出到 APT 系統。如以下匯出 Debian 公鑰的範例：

指令：

```
gpg --armor --export <金鑰指紋> | sudo apt-key add - # 匯出公鑰到 APT
```

將剛剛下載的 Debian 公鑰匯入 APT 系統：

```
dbtsai@ubuntu:~$ gpg --armor --export 010908312D230C5F | sudo apt-key add -
gpg: 沒有找到任何徹底信任的金鑰
OK
dbtsai@ubuntu:~$ sudo apt-key list
/etc/apt/trusted.gpg
```

```

-----
pub   1024D/437D05B5 2004-09-12
uid           Ubuntu Archive Automatic Signing Key <ftpmaster@ubuntu.com>sub
2048g/79164387 2004-09-12

pub   1024D/FBB75451 2004-12-30
uid           Ubuntu CD Image Automatic Signing Key <cdimage@ubuntu.com>

pub   1024D/2D230C5F 2006-01-03 [過期: 2007-02-07]
uid           Debian Archive Automatic Signing Key (2006) <ftpmaster@debian.org>

dbtsai@ubuntu:~$

```

若是在伺服器上找不到它的公鑰，讀者可以寫信去要，它會給你他的公鑰，大致上長得像以下範例一樣：

公鑰範例：

```

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.2.2 (GNU/Linux)

mQGIBEO6XBMRBACFyOjxsl7kkn0dnzRlMDHFZwcLR3A0xAcvC97jbmSvuiH2JlKu
RlJkFqCNGv3yzvtjflMRrNfmIgitOOaPmjK4erQoXM2cyRHlsk/OXLM2aGcR8PGE
略
-----END PGP PUBLIC KEY BLOCK-----

```

讀者把它存成文字檔 `debian.asc` 以後，可以透過以下指令匯入：

指令：

```
sudo apt-key add debian.asc
```

若是讀者要刪除公鑰，可以透過以下指令：

指令：

```
sudo apt-key del <金鑰指紋>
```


當然金鑰的使用不只在這裡，只要有關於安全性的部份，都常常見到它的縱影，往後筆者會在各種不同的應用領域再次介紹到，不過原理大致就如本節所敘述的。

5.3.5 系統升級與更動

版本	開發代號	釋出時間
4.10	warty	2004 年 10 月 20 日
5.04	hoary	2005 年 04 月 08 日
5.10	breezy	2005 年 10 月 13 日
6.06	dapper	2006 年 06 月 1 日
6.10	edgy	2006 年 10 月 26 日

在 Ubuntu 和 Debian 最為人稱道的就是套件升級的方便性，掌握了這些技巧，以後都不需要再重新安裝，就可以無痛一直升級下去！筆者這裡舉例，目前我的系統是 6.06 (dapper)，而 Ubuntu 已經出了最新版 6.10 (edgy)，若我想要更新到最新版，要怎麼辦呢？看完上面章節的讀者，大概已經知道要怎樣做了吧！不過這裡筆者不免要嘮叨一下升級時候需要注意的細節喔！

首先，用你最習慣的文字編輯器去編輯 `sources.list`

 指令：

```
sudo gedit /etc/apt/sources.list
```

 `sources.list` 修改前內容：

```
deb http://apt.ubuntu.org.tw/ubuntu/ dapper main restricted
deb-src http://apt.ubuntu.org.tw/ubuntu/ dapper main restricted

deb http://tw.archive.ubuntu.com/ubuntu/ dapper-updates main restricted
deb-src http://tw.archive.ubuntu.com/ubuntu/ dapper-updates main restricted
以下略
```

目前是使用 dapper 的套件庫，所以把所有 dapper 的地方改成 edgy。

 `sources.list` 修改後內容：

```
deb http://apt.ubuntu.org.tw/ubuntu/ edgy main restricted
deb-src http://apt.ubuntu.org.tw/ubuntu/ edgy main restricted

deb http://apt.ubuntu.org.tw/ubuntu/ edgy-updates main restricted
```

```
deb-src http://apt.ubuntu.org.tw/ubuntu/ edgy-updates main restricted
以下略
```

修改完了以後，緊接著執行下面兩行指令，使用 `dist-upgrade` 的原因是因為版本之間差異太大，需要考慮因相依性而裝新套件或移除套件的問題。

指令：

```
sudo apt-get update # 更新清單到 edgy 的套件庫
sudo apt-get dist-upgrade # 一定要用 dist-upgrade 喔！
```

若更新完後，不幸遇到錯誤，反覆的試試看以下 4 個指令，應該會有一些幫助哩！

指令：

```
sudo apt-get update # 更新套件清單
sudo apt-get -f -y upgrade # 遇到錯誤也繼續 upgrade
sudo apt-get -f -y dist-upgrade # 遇到錯誤也繼續 dist-upgrade
sudo dpkg --configure -a # 重新設定之前因中斷等還沒安裝好的套件
```


系統升級後，筆者建議檢查系統上重要的虛擬套件 (metapackage) 是否存在，若不存在的話，建議手動安裝上去。所謂的虛擬套件是實際上沒有內容的，它的作用只是用相依性來連結其他的套件。舉例來說，Ubuntu Gnome 桌面版最重要的 metapackage 就是 `ubuntu-desktop`，讀者可以用 `apt-cache show ubuntu-desktop` 去看它的相依性和套件大小。它的相依性有將近數百多個，所相依的這些套件都是桌面上 Gnome 應用時所需要先安裝的套件。

在系統完整的安裝完以後，metapackage 其實可以刪除，並不會造成你的系統有所缺陷。但重點是若在做重大更變時，如 `dapper` 到 `edgy`，有可能這些 metapackage 有增加新的相依套件，而當你的系統沒有這些 metapackage，會造成這些新套件不會被安裝到，因此新的功能也會減少。因此筆者認為 metapackage 應該要保留在系統上，雖然系統若一直維持在一個版本上沒什麼差別，但是在做系統版本升級時是很重要的。為了確定系統沒有因為不可知的原因把這些 metapackage 弄丟，筆者習慣是在系統版本升級後緊接著檢查一次 metapackage，若沒有裝上去的話，把它們都裝上吧！

表 5.3.1 常用的 metapackage 和語系套件


內容	套件名稱
Ubuntu Gnome 桌面 (metapackage)	ubuntu-desktop
Ubuntu KDE 桌面 (metapackage)	kubuntu-desktop
Ubuntu XFCE 桌面 (metapackage)	xubuntu-desktop
基本中文支援虛擬套件。相依輸入法以及 Firefox 和 OpenOffice 等的中文翻譯。(metapackage)	language-support-zh
zh_TW.UTF-8 和 zh_CN.UTF-8 的 locale 語系支援。(實體套件)	language-pack-zh
Ubuntu Gnome 桌面中文翻譯 (實體套件)	language-pack-gnome-zh
Ubuntu KDE 桌面中文翻譯 (實體套件)	language-pack-kde-zh

例如，筆者使用 Gnome 桌面，所以筆者會在 dist-upgrade 以後，執行以下指令來確認 ubuntu-desktop 是否存在。當然系統上若已經存在的話就會自動忽略。

 指令：

```
sudo apt-get install ubuntu-desktop # 安裝 Gnome 桌面的 metapackage
```

常常有人用 metapackage 的方式在 server 版上的 Ubuntu 裝視窗環境，但是使用者會發現中文不見了，也沒有輸入法，變成全英文版阿！這時候不用擔心，只要裝上中文支援的套件就可以解決啦！有時候在系統 dist-upgrade 前，中文的 metapackage 也會莫名其妙不見，導致升級後有些套件會缺中文翻譯，也可以透過這個方式解決。而這裡的範例是以 Gnome 中文系統為主喔！

 指令：


```
sudo apt-get install language-support-zh # 安裝常用中文支援的 metapackage
sudo apt-get install language-pack-zh # 安裝中文 locale 語系支援
sudo apt-get install language-pack-gnome-zh # 安裝 Gnome 中文翻譯
```

如果不是用 Gnome 的，可以如表 5.3.1 去察看要安裝哪個套件，若是使用多種桌面的，可以每個都去給它裝一下！

5.4 視窗模式的套件管理程式 Synaptic

Synaptic 是一個基於 APT 的前端視窗套件管理程式 (front-end)，因為它是基於 APT 所開發，所以幾乎 APT 系統裡面的所有功能，都可以直接用 Synaptic 來取代喔！更強大的是，它可以直接用滑鼠選擇要安裝的軟體，並且將軟體分類，找到想要安裝的軟體後，「標記為安裝」後再按「套用」，這時候它就會自動安裝下載，再也不用辛苦找套件名稱、再執行 `apt-get install`，因為它是 APT 的 front-end，所以它所使用的套件庫也是 `sources.list` 喔！

啟動的方式可以透過桌面上的選單，從系統→管理→Synaptic 套件管理程式來執行，或很簡單的在文字終端機下打以下指令：

 指令：

```
sudo synaptic
```

然後讀者可以看到如圖 5.4.1，讀者可以開始東玩玩西玩玩去認識它啦！它已經是用視窗圖形介面，所以不像用文字介面的套件管理系統一樣冷僻，應該讀者玩一玩就可以很快上手的。



圖 5.4.1

如同文字模式的 `apt-get update` 去更新套件庫的清單，Synaptic 相同的功能在面板上的「重新載入」，按了那個按鈕後，如圖 5.4.2，就會自動更新套件庫清單。

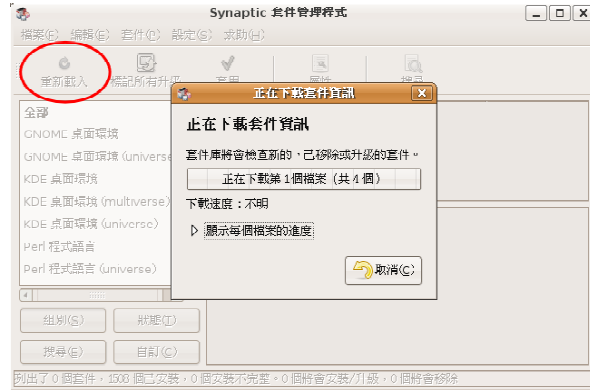


圖 5.4.2 更新套件庫清單

若要做同等於 `apt-get upgrade` 的系統更新，可以按面板上標記所有升級，它就會在各個需要升級的套件上給與標記，如圖 5.4.3。此時系統還沒有實際的升級，你還可以對其他要安裝或異動的套件作標記，最後在按下套用，如圖 5.4.4，那麼系統就升級好了，也會對你剛剛額外的異動與安裝作處理喔！

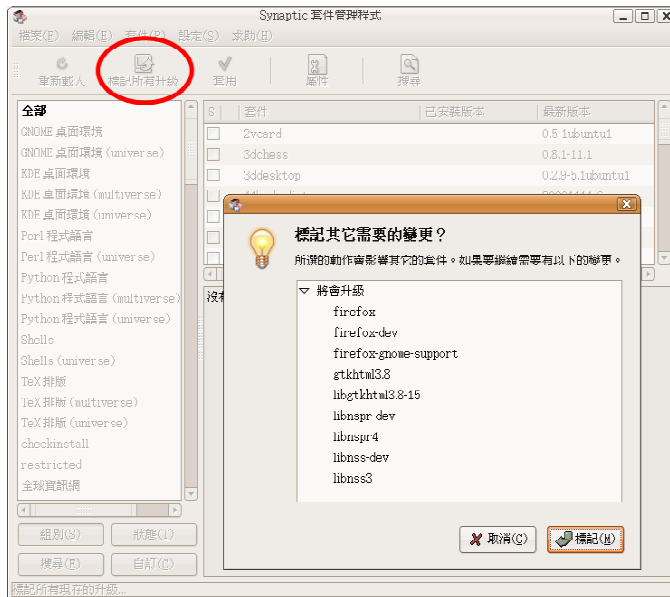


圖 5.4.3 標記所有升級

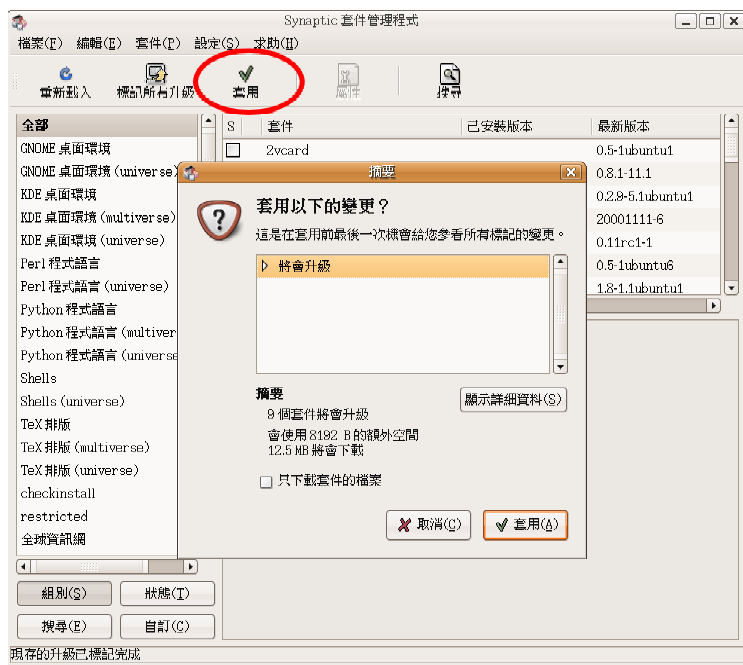


圖 5.4.4 套用剛剛所作的標記

而 Synaptic 有個特色，不論是你勾選了要安裝或要移除的套件後，它都只會把這些狀態標記起來，而不會選了以後就馬上執行。必須要如圖 5.4.4 按下「套用」才會真的安裝，移除或升級。這樣有幾個好處，讀者可以指定幾個操作後，再一次執行，效率上會比較好，並且最後還可以再作一次檢查，看看有沒有選錯的，然後再套用。

若要安裝軟體也很簡單，你可以從分類裡面去找你想要的軟體，也可以用搜尋的功能。筆者就舉個很簡單的例子，點選「編輯」裡面的「搜尋」，就可以用關鍵字尋找套件。找到你要的軟體後，在上面按著就可以如圖 5.4.5 選擇標記為安裝。你還可以繼續做一些安裝或移除的操作，最後確定了，按下「套用」就會幫你裝到系統上。

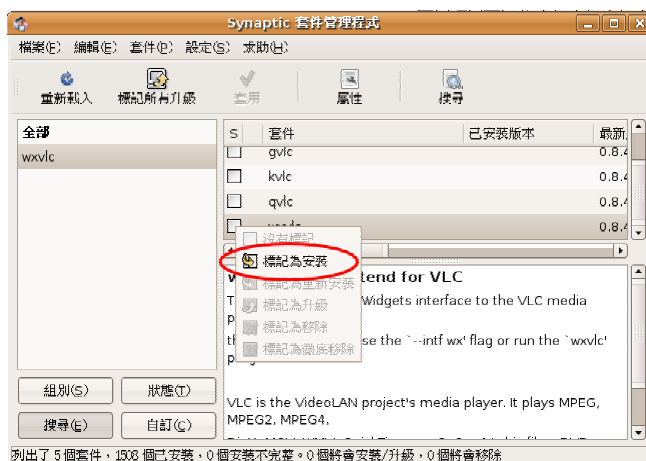


圖 5.4.5 標記 wxvlc 為安裝

用同樣的方法，也可以移除程式，在標記的時候也有 remove 或 purge 模式可以選擇，詳細方法，就讓讀者自己品味了喔！它有用 GUI 管理 sources.list 的功能，也有管理金鑰等選項，有興趣的讀者可以到「設定」裡面去瞧一瞧，應該是不會太難才是。

5.5 從原始碼建立套件

做為玩 GNU/Linux 的人，免不了要常常去把玩原始碼，自己從原始碼建立套件。原因或許是因為某個套件還沒有被打包進套件庫，也許是出了新版，但是套件庫還沒有更新，也有人是因為想要自己編譯套件取得最佳化執行效率。但不幸的是，這是非常枯燥煩瑣的過程，要自己注意編譯時的函式庫和標頭檔等問題，所幸這些問題都被 Debian 漂亮的解決了（Ubuntu 也繼承了這個特性 ^_^）。

若是一套軟體的原始碼有支援 Debian 套件管理系統的規則，我們稱為那套軟體已經 debianize。但是不可能每套軟體上游（作者所釋出）的原始碼都已經 debianize，所以 Debian 和 Ubuntu 都有專門的開發者幫這些自由軟體 debianize。debianize 的檔案呈現有點類似 RedHat 系列採用的 SRPM（Source RPM）方式，裡面除了原始碼以外，還包含很多額外的資訊，告訴套件管理系統在編譯前要先裝哪些標頭檔和套件，也寫著編譯好的套件，哪個檔案要裝到哪個位置等資訊。

若軟體原始碼是直接支援 Ubuntu (debianize) 的，軟體 tarball 解壓縮後會有一個 debian 的子目錄，那個目錄底下就放很多包裝套件時所需要的設定檔。

已經 Debianize 的 gftp 原始碼架構：

```
root@ubuntu:/home/build/deb/gftp/gftp-2.0.18# ls
ABOUT-NLS      config.guess    src             Makefile.in    README.html
acinclude.m4    config.h.in     debian         install-sh     mknstalldirs  THANKS
aclocal.m4      config.rpath    depcomp        intl           NEWS          TODO
ChangeLog       configure       gftp.spec      Makefile.am    README
ChangeLog-old   configure.in    gftp.spec.in

root@ubuntu:/home/build/deb/gftp/gftp-2.0.18# ls debian/
changelog  gftp-common.dirs  gftp-gtk.links  gftp-text.postinst
compat     gftp-common.docs  gftp-gtk.menu    gftp-text.preperm
control    gftp-common.install  gftp-text.install  rules
root@ubuntu:/home/build/deb/gftp/gftp-2.0.18#
```

當透過 apt-get source 取得軟體原始碼時會有 3 個檔案，一個是軟體上游的原始碼 (orig.tar.gz)，這可以是沒有 debianize，一個是修正檔 (diff.gz)，可以在上游原始碼加上 debianize 時所需的相關資訊，dsc 檔包含了對套件的描述和前兩個檔案的 MD5 hash 值。在 RedHat 裡面，這 3 個資訊是統一包在一個 SRPM 檔。

5.5.1 使用 apt-get 來從原始碼建立套件

最簡單取得已經 debianize 的套件原始碼就是透過 apt-get source。而且 APT 系統甚至能幫你做到安裝相關編譯所需套件。這個技巧筆者常常用在和開發版 Ubuntu 以及 Debian experiment 套件庫原始碼混用的情形。因為直接裝這些不是針對你系統所編譯的套件很容易造成系統損壞或者套件不能執行，所以這是讓你系統部份升級的最好方式唷！

筆者就以範例來解釋好了，筆者在寫這一節的時候，使用 Ubuntu dapper 裡面所包含的 inkscape 0.43 版。這套軟體是 Linux 底下的向量繪圖自由軟體，本書很多圖都是用它畫的呢！目前最新的版本是 0.44，它有支援筆者夢寐以求的圖層功能。這時候筆者是可以抓 tarball 下來編譯安裝，但是這樣就無法利用套件管理系統來管理。所以筆者要把它從原始碼編譯成二進位 Ubuntu 套件，這樣套件管

理系統就可以控管了。此時筆者又發現 Ubuntu Edgy 裡面有包含 0.44 版，但是它的二進位檔因為編譯環境不同所以無法在筆者目前的系統運作，因為此時 edgy 還很不穩定，筆者也不想把整個系統升級到 edgy。於是筆者把 sources.list 裡面的 deb-src 改成 edgy 的，打算從這裡取得已經 debianize 的原始碼來建立套件。這種技巧也可以拿來用在編譯安裝 Debian deb-src 的套件，尤其是在 Debian 的 experiment 裡面有很多超新的玩意可以玩 ^_^。

 範例，在 6.06 dapper 底下編譯 6.10 edgy 的 inkspace 套件：

```
dbtsai@ubuntu:~/temp$ sudo gedit /etc/apt/sources.list # 把 deb-src 裡面的 dapper
改成 edgy
dbtsai@ubuntu:~/temp$ sudo apt-get update # 更新套件清單
dbtsai@ubuntu:~/temp$ sudo apt-get build-dep inkscape # 自動安裝編譯 inkscape
所需套件
dbtsai@ubuntu:~/temp$ sudo apt-get -b source inkscape # 下載原始碼並且編譯
# 此時若沒有錯誤，應該已經開始編譯了，這時候可能要一陣子時間，
# 你可以繼續讀本書下去，或去喝個咖啡休息！
dbtsai@ubuntu:~/temp$ ls
inkscape-0.44 inkscape_0.44-1_powerpc.changes
inkscape_0.44-1.diff.gz inkscape_0.44-1_powerpc.deb
inkscape_0.44-1.dsc inkscape_0.44.orig.tar.gz
dbtsai@ubuntu:~/temp$ sudo dpkg -i inkscape_0.44-1_powerpc.deb #安裝剛做好套件
```

若你在 apt-get build-dep 時出現錯誤，可能是因為該套件編譯相依的表頭檔和函式庫比較新，而這些編譯相依套件是從 deb 這個套件源下載安裝，因此會造成找不到新版的相依套件而無法滿足編譯所需的相依性。若你的 deb 和 deb-src 都是同一個版本，就一定不會出現這種錯誤，就算是不同，出現錯誤的機率也不高。這個時候可以先略過這個步驟，先取得原始碼，然後再稍微手動修改編譯相依性規則。實際上 inkscape 0.44 用以上的方法是編譯不起來的，會遇到這裡所說的問題，不過大部份軟體用上面的方式都可以成功建立套件。

以下是示範當上面的招術行不通時，用手動來做細部 hack 修正的方法。

 範例，inkscape 編譯相依性修正

```
dbtsai@ubuntu:~/temp$ apt-get source inkscape # 下載原始碼
# 有 -b 的參數代表了系統會在下載完原始碼後自動編譯，沒有的話只會下載原始碼。
dbtsai@ubuntu:~/temp$ ls -l
```


```

總計 9367
drwxr-xr-x 9 dbtsai dbtsai    2032 2006-08-04 19:44 inkscape-0.44
-rw-r--r-- 1 dbtsai dbtsai   22041 2006-06-28 01:17 inkscape_0.44-1.diff.gz
-rw-r--r-- 1 dbtsai dbtsai     948 2006-06-28 01:17 inkscape_0.44-1.dsc
-rw-r--r-- 1 dbtsai dbtsai 9549500 2006-06-28 01:17 inkscape_0.44.orig.tar.gz
# 記不記得一個 debianize 的原始碼套件要有 3 個檔案呢? diff.gz, dsc, orig.tar.gz
# 而使用 apt-get source 會自動把 orig.tar.gz 解開, 並且用 patch 把 diff 打上去。
# 解開原始碼且打上 patch 後產生的目錄 inkscape-0.44 我們通常稱為 build tree。
# 若是你是直接在網路上取得這 3 個檔案, 你可以用
# dpkg-source -x inkscape_0.44-1.dsc 來把檔案解開到 inkscape-0.44
# 若你連 dsc 檔都沒有, 那可以 tar xvf inkscape_0.44.orig.tar.gz 解開原始碼,
# 再用 zcat inkscape_0.44-1.diff.gz | patch -p0 來打上 debianize patch!
# 最後那個打 patch 的動作實際上就是在原始碼目錄下產生 debian 這個子目錄,
# 而用處就是拿來紀錄包裝套件的資訊。
dbtsai@ubuntu:~/temp$ cd inkscape-0.44/
dbtsai@ubuntu:~/temp/inkscape-0.44$ dpkg-checkbuilddeps # 檢查需要的函式庫
dpkg-checkbuilddeps: Unmet build dependencies: libart-2.0-dev (>= 2.3.10)
libgc-dev (>= 1:6.7-1) libglib2.0-dev libgnomevfs2-dev libgtk2.0-dev (>= 2.0.6-1)
libgtkmm-2.4-dev liblcms1-dev libpangol.0-dev libperl-dev libpng12-dev
libpopt-dev libsigc++-2.0-dev (>= 2.0.16-2)
# 或者你可以直接看 debian/control 裡面的 Build-Depends, 並把這些套件補齊!
# 也可以用 apt-cache showsrc <套件名稱> 來取得編譯所需的套件名稱。
dbtsai@ubuntu:~/temp/inkscape-0.44$ sudo apt-get install <所欠缺的套件>
# 這裡可以把上面檢查的結果直接拷貝貼上, 記得要把版本的地方去掉。
dbtsai@ubuntu:~/temp/inkscape-0.44$ dpkg-buildpackage
# 此時你會出現錯誤, 系統抱怨 libgc-dev 不夠新, 沒關係, 只要把 debian/control
# 裡面 libgc-dev 後面版本指定的地方改小一點, 或者乾脆砍了! 然後在重新一次
# dpkg-buildpackage, 等編譯完成後, 應該在上一層目錄有 deb 套件檔, 再用
# dpkg -i 安裝即可。

```


當然你可以在修改 deb-src 到其他版本以前, 先做 apt-get build-dep 的動作, 這樣一定不會出現錯誤訊息。然後再修改到最新的 deb-src 後取得原始碼, 再來試試看 dpkg-buildpackage 缺什麼, 或者哪裡版本不合需要修改 debian/control。通常這樣會比較少編譯相關套件需要手動安裝。

若是你的系統之前沒有用這種方式建立套件過, 而且第一次用 apt-get build-dep 就失敗, 那麼需要先手動安裝基本的編譯套件, 底下指令會幫你安裝 gcc compiler, make 和 dpkg-dev 等基本的程式開發環境。


 指令：

```
sudo apt-get install build-essential
```

為了讓指令比較容易查詢，底下筆者簡單的把所用過的指令個別列出來一次，而詳細的用法，讀者可以在範例中了解。

 取得軟體原始碼：


```
sudo apt-get source <套件名稱> # 可以加上 -b 在取得後編譯。
```

 安裝編譯此套件所需要的套件（通常是表頭檔或函式庫）：

```
sudo apt-get build-dep <套件名稱>
```

 建立 debianize 的原始檔方式：

```
dpkg-source -x <dsc 檔案名稱> # 解開套件原始碼並且打 patch
# 這動作共需要 dsc, orig.tar.gz 和 diff.gz 3 個檔案。它先把 orig.tar.gz
# 解開後在打上 diff.gz 這個 patch
```

 當沒有 dsc 檔時，建立 debianize 的原始檔方式：

```
tar xvf <orig.tar.gz 原始檔檔名> # 解開套件原始碼
zcat <diff.gz patch 檔檔名> | patch -p0 # 打上 debianize 的 patch
```

 開始編譯套件：

```
dpkg-buildpackage # 必須在軟體原始碼的目錄下執行喔！
# 加上 -rfakeroot 可以不用 sudo 或用 root 編譯，但需先安裝 fakeroot 這個套件
# 加上 -uc 可以不在套件上簽名，-b 只產生套件檔，不會在把原始碼打包一次。
```

5.5.2 使用 apt-build 來建立最佳化套件

有些人或許會羨慕 gentoo 或者 BSD 系列作業系統，它允許使用者可以對系統上的套件做最佳化。而 Ubuntu 和 Debian 為了機器的相容性，預設套件是對 368 編譯，這多多少少也損失了一些效能，比較起來大致上會比 gentoo 這種對機器極度最佳化的 Linux 發行版，大概會慢個 10% 到 20%。

但是讀者別失望，Ubuntu 和 Debian 其實有提供讓使用者自行編譯最佳化的功能，由於編譯時間費時許久，筆者建議是用在關鍵的套件上。例如常上網的可以

最佳化 FireFox，或者在伺服器使用可以最佳化 Apache 等。這裡要提醒讀者的是，有時候自訂的最佳化程度弄的太過火的話，會讓程式無法編譯或無法執行喔！

首先要安裝 apt-build，透過這個程式來達成編譯套件最佳化！

指令：

```
sudo apt-get install apt-build
```

安裝的時候，會進入設定畫面，而第一個設定選項設定是原始碼下載位置，選擇預設的位置即可。第二個設定是編譯後套件要放在哪裡，這也採用預設即可。然後系統會問題你所需要的最佳化程度，建議是選 Medium，因為有時候 Strong 會出現穩定性問題。再下一個選項是要不要把編譯後的套件放入本地端的套件源內，當然選擇 yes，除非你要手動安裝最佳化後的軟體。最後它會問你系統的平台，你可以選擇符合你機器的平台。若是沒有的話，可以選擇次一級的，那沒有關係。

若是你想要重新設定 apt-build 選項，可以透過以下指令：

指令：


```
sudo dpkg-reconfigure apt-build
```

當然你也可以直接修改設定檔，其位置在 `/etc/apt/apt-build.conf`

修改 apt-build 設定檔：


```
build-dir = /var/cache/apt-build/build           # 下載的原始檔和編譯位置
repository-dir = /var/cache/apt-build/repository # 編譯好套件位置
Olevel = -O2                                     # 編譯最佳化程度， -O3 是最強的，但可能不穩定
mtune = -mtune=athlon64                          # 這個值和你的架構有關，你可以藉由重設 apt-build
options =                                         # 來改變，或在網路查詢。
make_options = " "
dbtsai@ubuntu:~$
```

不免俗的，首先要更新套件清單。

 指令：


```
sudo apt-build update
```

若你要裝一個新套件，且最佳化編譯，可以透過以下指令：

 指令：


```
sudo apt-build install <套件名稱>
```

若是已經安裝過的套件，需要重新最佳化編譯，可以透過以下指令：

 指令：

```
sudo apt-build --reinstall install <套件名稱>
```

apt-build 系統也有提供類似 BSD 那種整個系統重新編譯，不過這不一定能成功，也可能導致系統損壞。如果有興趣的讀者可以嘗試看看。

 指令：

```
sudo dpkg --get-selections | awk '{if ($2 == "install") print $1}' >
/etc/apt/apt-build.list
# 首先把系統內所有安裝的套件清單匯出，這些套件將會重新編譯
sudo apt-build world          # 開始全系統編譯，應該需要耗個幾天 ^_^
```

筆者要提醒你的是，這些自己編譯套件和最佳化的技巧是不一定能成功，有可能會毀壞你的系統，如果你要嘗試的話，需要自己承擔後果喔！


5.5.3 使用 auto-apt 來自動滿足 tarball 相依性

對於那些沒有 debianize 的程式，若我們要手動從 tarball 編譯，往往是將軟體原始碼解開後，使用以下方式編譯並且安裝。

```
./configure          # 設定軟體相關資訊並檢查是否滿足編譯條件
make                 # 開始編譯
sudo make install    # 安裝到系統
```


但是常常會在 configure 的時候，碰到缺少編譯相關套件的問題。且不幸的是，常常它顯示出來缺少的套件會和 Ubuntu 命名不同。所以光是在補這些東西就要耗掉很多時間了。

此時可以透過 auto-apt 來自動安裝從 tarball 編譯軟體所需要的套件。

 指令：


```
sudo apt-get install auto-apt
```

更新 auto-apt 所使用的清單

 指令：

```
sudo auto-apt update
```

這個時候你可以用以下方法來 configure，若遇到缺少的套件，它會另外跳出一個視窗問你是否需要安裝。

 範例：


```
auto-apt run ./configure # 若 configure 遇到缺少的套件，會自動詢問要不要裝  
make  
sudo make install
```

讀者也可以配合下一小節的 checkinstall 來產生 deb 套件檔。

5.5.4 使用 checkinstall 來從 tarball 建立套件

要把 tarball 的原始碼徹底 debianize 其實蠻麻煩的，需要先自動產生一些套件資訊樣版，然後還要再細部修改。所以這裡筆者不談這種方式，改成教大家使用比較簡單的 deb 套件檔包裝方式。當然一個套件若要進入官方就得要先 debianize，也是把套件包成 deb 的最正統方式，若讀者有興趣，可以到網路上尋找 HOWTO。

首先需要先安裝 check install

 指令：

```
sudo apt-get install checkinstall
```

這裡筆者舉還沒有被打包進去套件庫的 FileZilla 3.0CVS beta 來作為範例。首先在網路上取得最新版的原始碼，並且將它解壓縮。如範例，剛開始從傳統的 configure 開始，若這裡若發生錯誤，可以配合 auto-apt 來安裝編譯所需套件。

範例：

```

dbtsai@ubuntu:~/filezilla-3.0cvs20060708$ ./configure
dbtsai@ubuntu:~/filezilla-3.0cvs20060708$ make                # 開始編譯
dbtsai@ubuntu:~/filezilla-3.0cvs20060708$ checkinstall
checkinstall 1.5.3, Copyright 2001 Felipe Eduardo Sanchez Diaz Duran
    This software is released under the GNU GPL.

The package documentation directory ./doc-pak does not exist.
Should I create a default set of package docs?  [y]: n
# 這裡問我們要不要建立文件的備份，一般是不用

Please write a description for the package.
End your description with an empty line or EOF.
>> Filezilla ftp client
>>
# 這裡可以讓我們輸入對於軟體的簡單說明

This package will be built according to these values:

0 - Maintainer: [ root@ubuntu ]
1 - Summary: [ Filezilla ftp client ]
2 - Name: [ filezilla-3.0cvs20060708 ]
3 - Version: [ 3.0cvs20060708 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ checkinstall ]
7 - Architecture: [ powerpc ]
8 - Source location: [ filezilla-3.0cvs20060708 ]
9 - Alternate source location: [ ]

Enter a number to change any of them or press ENTER to continue:
# 這裡可以選擇數字來更改套件相關資訊，若不要更改就按 ENTER

```

```
dbtsai@ubuntu:~/filezilla-3.0cvs20060708$ ls *.deb
filezilla-3.0cvs20060708_3.0cvs20060708-1_powerpc.deb
# 終於順利產生 deb 套件檔啦！若要安裝就用 dpkg -i 哩！
```

如範例，整個步驟和傳統上使用 tarball 裝軟體差不多，只不過最後的 `make install` 改成 `checkinstall` 而已唷！

CHAPTER 6

系統設定與效能調校

大家安裝好了系統後，一定會想要先把常用的系統環境設定好，這樣才會有舒適的工作環境。所以本章會介紹一些常用的硬體驅動設定和調整，包含了網路設定、顯示卡設定、印表機安裝和終端機外觀的調整，甚至教大家如何開啟硬碟的 DMA 模式來加速等技巧。讀者可以照著範例 step by step 的操作。

6.1 網路設定

筆者認為玩 Linux 就一定要有網路，如果沒有網路的話，Linux 就一點都不好玩啦！為什麼呢？例如 Ubuntu 最強大的 APT 套件管理系統就要有網路的配合，或者 Linux 最擅長的伺服器領域不是也和網路扯上關係嘛！所以網路是 Ubuntu 不可以或缺的功能呢！

初學 Linux 的人可能還對網路架構不熟悉，而這些網路的基本知識對於往後的學習有一定的重要性，所以筆者會先稍微介紹一下網路概要。但這不是一本專門講網路架構的書，所以筆者只會探討到那些學習本書所需要的預備知識。

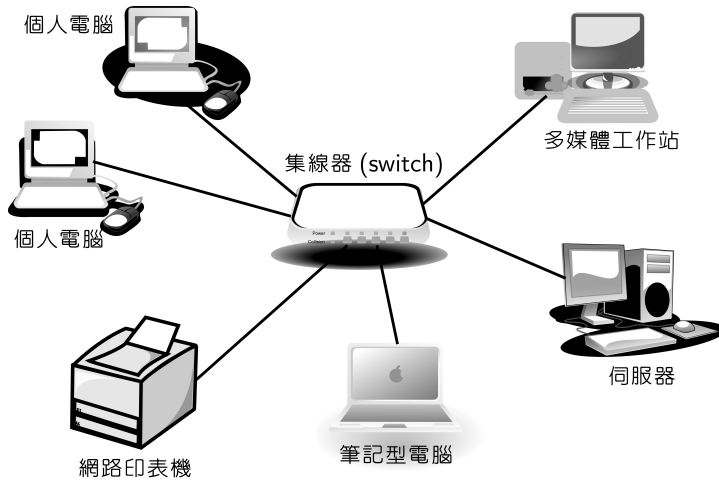
接下來才會開始教讀者如何實際設定網路等。由於很多人是把 Linux 當成伺服器來使用，這時候常常不會安裝圖形介面，所以在指令列操作網路設定還是有存在的必要。因此筆者把視窗介面設定網路和文字介面的設定方法分開來，讀者可以視需要來研讀！

6.1.1 網路概論

不同的區域網路架構

早期區域網路 (Local Area Network, LAN) 的架構是沒有統一的標準，造成了各家的網路產品只能和自家的產品溝通，那時候也不盛行網際網路，透過標準的 TCP/IP 協定來橋接不同的網路架構，這使得網路推廣產生了很大的阻力。後來全錄 (Xerox)、英特爾 (Intel) 和迪吉多 (Digital) 共同推出了乙太網路 (Ethernet) 這種區域網路規格，由於標準化且便宜，因此迅速的流行起來！這時候大家常在 Ethernet 上跑的網路協定還是專有規格，如 Novell 的 IPX 協定等，不是 TCP/IP。雖然這種使用模式不能跨網路架構溝通，但因為大家都用乙太網路，所以也沒有跨架構的煩惱存在！

因此個人電腦目前常用的區域網路架構，就是乙太網路！常用的乙太網路實體電路架構如圖 6.1，我們常常把各種不同的設備接到 switch 或者 hub 上，形成星狀結構。當然若是 switch 插滿了，我們還可以用另外一個 switch 接上去，這樣就形成另外一個星狀架構。



乙太網路星狀拓撲 (區域網路)

圖 6.1

那資料是怎樣在乙太網路上傳遞？很簡單，若有一台電腦要傳送資訊給同一個區域網路上的另外一台電腦，它就會向所有區域網路上的電腦廣播 (broadcast)，而廣播的資料封包帶有 MAC (Media Access Control) Address，這個 Address 是每個乙太網路卡獨一無二的識別碼，因此區域網路上的電腦可以根據封包的 MAC address 來判斷是否要接收，若不是自己的 MAC address，就把這個封包丟棄。這個 MAC address 有 6byte，前 3byte 是和網卡公司有關，後 3byte 是網卡生產的序號，這樣的 MAC 值理論上全世界只有一個喔！

但是日前有某家主機板廠商很糊塗，生產的主機版上內建網卡竟然用同一個 MAC 值，這會怎麼樣呢？若是在同一個區域網路底下 (同一個 switch 或 hub)，當同 MAC 網卡的機器一起工作時，會造成不知道封包要不要接收的情況吶！這時候網路就會開始不正常啦！當然若是不在同一個區域網路，但都在網際網路上，這樣就沒有關係，因為網卡編號只用來在區域網路對傳使用喔！且封包上網卡編號是放在 TCP/IP 第一層網路存取層，過了 router 後，MAC address 就會變成 router 的網卡啦！

因為採用廣播的方式傳遞資訊，在早期使用 hub 時，若太多人同時廣播就會產生資料碰撞 (collision)。為了減少這種情形的發生，乙太網路使用了一種稱為 CSMA/CD 的技術，此時當乙太網路卡要傳送資訊出去，會先聽網路上有沒有人

在廣播，若是沒有的話才會把資訊送出。若每次可以傳輸的封包太大，這樣子別台電腦要等很久，因此乙太網路規定每個封包最大只能有 1500byte，傳輸完了就該換人而不能一直霸佔著頻寬。這個值就是所謂的最大傳輸單位 MTU (Maximum Transmission Unit)。

其實乙太網路卡真正傳輸的實體框架 (frame)，其實有 1526byte，多出來的 26byte 是放著在 TCP/IP 四層架構中的網路存取層表頭，這裡面紀錄了來源主機的 MAC，目的地主機的 MAC 等。整個 frame 架構可以用圖 6.2 來表示。所以乙太網路卡每送出 1526byte 真正能用的就只有後面的 1500byte，這裡面封裝的資料可能是 TCP/IP 網路層封包。當然放在網路層的資訊不一定是我們熟悉的 TCP/IP，也有可能是 IPX 等網路協定哩！

前言 8byte	目的 MAC 6byte	來源 MAC 6byte	封包 類型 2byte	包裝網路層封包 46byte 到 1500byte (1500byte 就是所謂的 MTU)	FCS 4byte
-------------	--------------------	--------------------	-------------------	--	--------------

乙太網路實體連結層 Frame 圖

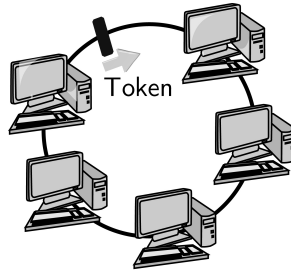
圖 6.2

想想每次要傳輸資訊時就得停下來等網路沒人用了才能使用，這是多麼沒有效率的事情啊！於是就誕生了 switch 這種東西來取代早期的 hub。switch 它可以自動紀錄哪個插孔是對應哪個目的地 MAC address，當封包送出去後，就只送到對應的 port，不需要廣播。因此不需要停下來等整段網路廣播完畢才能傳輸資訊，故可以想成 switch 每個 port 都有獨立的頻寬。

聰明的讀者一定會想，反正現在不用等別人廣播完畢，這樣為何不把 MTU 加大，如此浪費在表頭封包的比例就會比較少，而且傳輸時檔案就不用切割那麼多份，這樣不是有效率多了嘛！的確如此，現在有些 Giga ethernet 有支援 Jumbo frame，若配合有支援的 switch 最大可以把 MTU 開到 16000byte 呢！MTU 變大對於網路效能有很大的影響，尤其當傳輸大檔案時不用切割成那麼多片，然後再重組，也少了 overhead 的問題，這時候會發現效能有特別明顯的改變呢！

當然市面上不是只有乙太網路這種 LAN 架構，還有 IBM 的 Token Ring 等。如圖 6.3，它的實體電路是把所有電腦串成一個圓圈，第一台電腦先取得發言權！擁有發言權的電腦稱為拿到 Token。這些拿到 Token 的電腦發言時，會把要說的

話和目的地 MAC 寫在 Token Packet 上，然後把發言權傳給下一位。這一位剛拿到發言權的人會先檢查 Token Packet 上有沒有要給它的封包，有就讀取出來並清空寫給它的內容。之後再把想要說的話寫到 Token packet 上，再傳給下一位。因為它是圓圈狀的拓撲，所以最慢傳一圈就可以收到了。它只有拿到 Token 的人可以講話，且是一個接一個傳遞，故沒有碰撞的問題，所以 Token Packet 越大越好，不然給幾個人發言後寫滿了不就沒得玩了！因此 16 Mbit/sec Token Ring LAN 的 MTU 有 17914byte 那麼大呢！大約比以太網路的 MTU 大十倍以上。



Token Ring 區域網路架構

圖 6.3

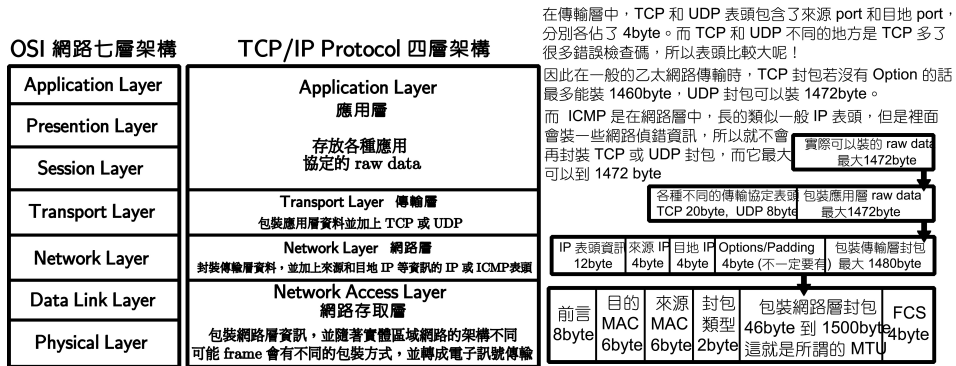
不同架構的區域網路要如何溝通呢？舉例來說，Ethernet 要如何和 Token Ring 溝通呢？它們連 MTU 大小都不一樣了，那怎能傳遞實體 frame 框架？這時候就要靠 TCP/IP 這種網路協定啦！由於 TCP/IP 設計的時候就定位為網際網路（Internet）使用，所以有考慮到封包會經過不同的區域網路要如何處理的議題。

網路分層概念

談到 TCP/IP 架構，不免俗的就一定會談到 OSI 七層網路模型，或者 TCP/IP 的四層網路架構。所謂的架構是把網路處理的細節抽象化，然後整理出來的一套普適的模型，讓我們很簡單的了解網路溝通的分工流程。當初 OSI 七層模型是為了建立出一個網路溝通的標準所定立，但在真正開發通訊協定時，它可以是一個參考，不一定真的需要把架構拆成七個。

或許讀者又會覺得奇怪，為什麼要把網路拆成那麼多層，一層或兩層不就簡單多了嘛！假設現在網路只有一層，大家都直接在網路存取層工作，那麼每套應用程式都要親自去處理很低階的硬體封包行為，甚至還要自己處理封包轉送的問

題，如果遇到不同的 LAN 架構，應用程式甚至要自己上火線處理 MTU 的問題！若像 TCP/IP 這樣拆成四層時，應用程式只要把資料餵給傳輸層，其他的事情都可以不用管啦！因為底下的網路層和更底下很低階的實體連接層都是作業系統和硬體直接管理，這時候不同架構 LAN 的特性問題就會在網路層被作業系統處理，來適應實體連接層所能裝的封包大小！這樣對於寫一般應用軟體的人不就輕鬆多了，只需要顧好應用層的資料和傳輸層要給與的表頭即可！



TCP/IP 網路封包架構示意圖

圖 6.4

從圖 6.4 可以很清楚的了解到 TCP/IP 四層架構和 OSI 七層模型對照的部位。如在 TCP/IP 架構把 OSI 第五層到第七層整合成應用層，並且把第一和第二層整合成網路存取層看待。讀者可以很清楚的從封包的包裝示意圖了解到，在 TCP/IP 架構中封包是層層封裝的，也就是封包都會放到下一層包裝起來。在實體電路傳遞的是最下面那一層，當電路把資料傳遞到電腦後，電腦再逐層解開，判讀該層的表頭資訊以便做下一步的處理！大致上可以把這四層很簡略的用以下敘述說明：

- TCP/IP 第四層：應用層（Application Layer）：裡面直接放各種應用軟體的原始資料，可能是網頁，檔案或者 FTP 等資訊。然後這些原始的資料會直接放在下一層傳輸層包裝。
- TCP/IP 第三層：傳輸層（Transport Layer）：第四層資料包裝進來後，它會在表頭上視封包用途和種類加一些料！若是 TCP 表頭的話就會加上目標 port 和來源 port 以及一些錯誤檢查碼等。若是 UDP 的話，就加上目

標 port 和來源 port 等表頭。因為 port 的資訊在這裡，所以我們用 iptables 擋 port 就在這一層運作喔！

- **TCP/IP 第二層：網路層（Network Layer）**：它把傳輸層的資料包進來以後，還會加上 IP 表頭。所謂的 IP 表頭就包含了來源端和目的地端的網路地址，上面也有一些欄位可以紀錄路由位置和整個封包大小。還有一種特別的表頭是 ICMP，它也有 IP 資訊，但多了一些網路偵錯用欄位，並且上面不再放 UDP 或 TCP 資訊。這一層包裝起來的封包大小最大不能超過 MTU，若超過就會切割成比較小份，好讓我們等下把這層封包包裝到實體電路中框架（frame）來傳遞。也因為這裡有 IP 資訊，所以路由決策是在這一層工作，所以最陽春的路由器只要處理到這一層就好了。
- **TCP/IP 第一層：網路存取層（Network Access Layer）**：把第二層包進來後，視該架構的實體電路設計和限制，加上該 LAN 特有的 frame 表頭，再送到電路上傳遞。例如 Ethernet 就會加上 26 byte 的 frame 表頭在這一層，裡頭描述著在區網中資料要傳遞到哪裡！也因為這一層獨立出來，才能夠很容易的透過 TCP/IP 跨 LAN 規格傳遞資料，僅要做的不過是在不同 LAN 架構上實作不同的 frame 即可。所謂的 frame 是在各個區域網路內進行物理傳送的資料框包，所以每一種不同區域網路的 frame 長的不一樣，但是裡面包裝的網路層 packet 資料格式是一樣的喔！

從這架構上我們可以很容易猜出，當封包流入電腦時，是先看第一層表頭 MAC address 是不是要送給本機網路卡的，若是的話就接收，若不是就把它丟棄！然後我們會看第二層網路層表頭，若是本機 IP 就接收，若不是的話就送到預設的路由器裡！到了第三層，我們會從表頭知道封包種類，若是 TCP 或 UDP，就會依照目的地 port 送給相對應的應用軟體。假使我們要從本機把資料送出去，那應用軟體會從第四層把資料丟出來，然後經過層層包裝，包第一層後，再經由實體線路傳輸。

由以上討論我們知道 port（通訊埠）就是實作在傳輸層上，可以想成 IP 是你家的門牌，而 port 可以是不同樓層。一般不同的應用軟體會使用不同的預設 port，表 6-1 和表 6-2 分別列出了 TCP 和 UDP 常用的 port。

表 6-1 常用的 TCP port

服務名稱	TCP port	說明
FTP	21	FTP 命令通道
SSH2	22	加密登入用
Telnet	23	BBS 或遠端登入用
SMTP	25	E-Mail 發送
DNS	53	網域名稱查詢
WWW	80	網頁瀏覽用
POP3	110	下載 E-Mail
HTTPS	443	加密的網頁協定

表 6-2 常用的 UDP port

服務名稱	UDP port	說明
DNS	21	網域名稱查詢
DHCP Server	22	自動配置網路組態伺服器
DHCP Client	23	自動配置網路組態客戶端

Router 與 Gateway 概念

先前談過，當資料被封裝到網路存取層時，這些資料只能透過 LAN 架構傳遞。但若要跨不同架構的時候，該怎麼辦？這個時候就可以看到 TCP/IP 分層結構的好處。當本機要傳遞資訊到不同網域或架構時，我們會透過路由器（router）來達成傳遞。通常路由器會有兩塊網路卡，兩塊網路卡分別在不同的區域網路。當本機要送出資料到網域外時，本機會在網路存取層的表頭填上該路由器和本機同區網的網卡 MAC 值，但是該封包的網路層表頭卻不是該網路卡的 IP Address。這時候網路存取層一定會把該框架送到此路由器，但路由器收到後，解開到網路層時，會發現這不是要給自己 IP 的阿！所以它就會把該封包照著路由規則送到另外一個網域的網卡，並在那裡重新包裝成該區網架構的第一層框架，這時候 frame 的來源 MAC 值就會變成路由器的，所以 MAC address 是無法跨網域的。

故當兩端區網架構不同時，例如封包從 MTU 比較大的 LAN 傳到比較小的 LAN 時，整個流程是這樣：frame 先在比較大 MTU LAN 那端 router 網卡，解開封包後，router 在網路層發現這不是給它的，需要再轉送出去。此時又會發現，目的地端 MTU 比較小，這時候就會在網路層切割（fragment）這些封包，好讓它的大小有辦法放進去目的地端的網路存取層。當然若從比較小的 MTU LAN 丟資料到比較大 MTU LAN 就不需要切割，就可以直接到達囉！

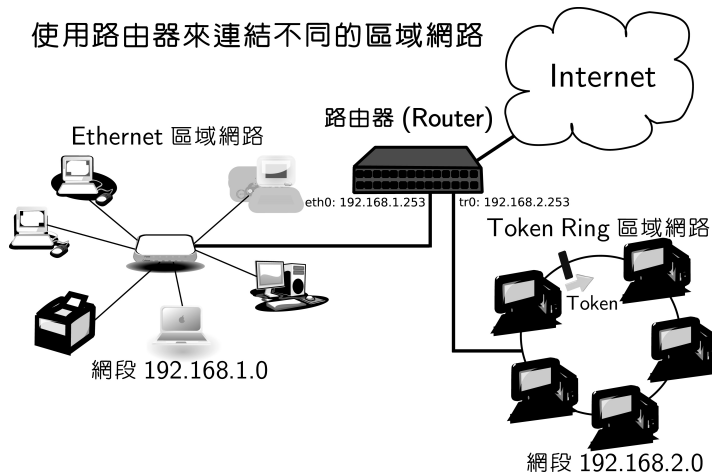


圖 6.5

網路遮罩（Netmask）概念

那麼本機網段的電腦要怎樣知道 B 網段電腦的 IP 不是屬於自己的網段，需要轉交給 router 處理？這時候就是依賴於網路遮罩（netmask）來決定。每個電腦都會設定它網域的 netmask，和閘道器（gateway）。所謂的 gateway，通常是 router 靠近你網域那一端的網卡 IP。當本機要送出封包時，它首先會把目的地 IP 和自己網路的 netmask 做運算，若得知是非自己網段的目的地 IP，就把資料往路由器（gateway IP）送，若是同網域的，就使用區網傳遞的機制處理。因此整個 Internet 是很多個區域網路透過路由器連接起來的，當路由器收到不是給自己網域的封包時，就會透過路由規則繼續往其他網域寄送。

備註 目前常用的 IP 格式是 IPv4 (Version 4 of the Internet Protocol)。IP 其實是一個 32 位元的整數，但為了方便記憶，我們把它拆成 4 個 8 位元的區間，而每個 8 位元以十進位來表示可以從 0 到 255，所以我們的 IP 就會長得像這樣 192.168.1.25。

那 netmask 要怎樣和目地 IP 做運算？若兩端的 netmask 都是 255.255.255.0，運算時本機 netmask 和目地 IP 會做二進位 AND 邏輯運算，算出來的結果就是目地網段的代號，於是可以透過該代號判別是否屬於自己網段。詳細的計算方式可以看圖 6.6 的範例。

網路組態：來源 IP 192.168.1.20 netmask 255.255.255.0 gateway 192.168.1.253

範例一：連線到同網域

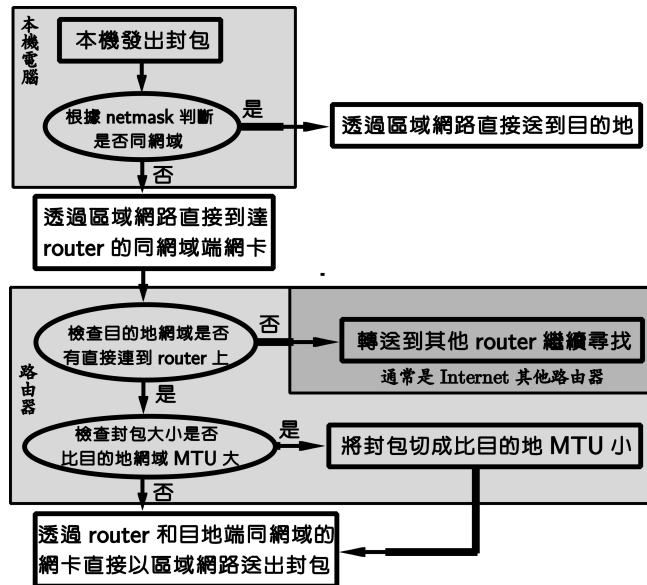
目地 IP A	:192.168.1.12	二進位	11000000.10101000.00000001.00001100
來源 netmask:	255.255.255.0	二進位	11111111.11111111.11111111.00000000
— AND 邏輯運算 —			
運算結果	192.168.1.0	二進位	11000000.10101000.00000001.00000000
	↑		
判別為同網域，可直接透過區域網路連結			

範例二：連線到不同網域

目地 IP A	:192.168.2.41	二進位	11000000.10101000.00000010.00101001
來源 netmask:	255.255.255.0	二進位	11111111.11111111.11111111.00000000
— AND 邏輯運算 —			
運算結果	192.168.2.0	二進位	11000000.10101000.00000010.00000000
	↑		
判別為不同網域，需轉交 gateway 192.168.1.253 透過 router 轉送封包			

netmask 計算示意圖

圖 6.6



封包在 Internet 傳送路由示意圖

圖 6.7

由於 IP 在配發的時候有分 Class A, Class B 和 Class C 這三網段，每一種的預設 netmask 都不一樣。Class A 是很大的網段，netmask 預設是 255.0.0.0，也就是總共可以有 $256^3 = 16777216$ 個 IP。不透過路由器，這麼大的網段會造成效能不好，所以常會用 netmask 的觀念切割成子網路！由於現在 IP 不夠用了，這種技巧也常常用在學術單位，學校常把一個 Class C 網段的 256 個 IP 拆成數個子網路。整個概念和目的地 IP 和本地 netmask 比較相同，其實當我們決定了哪些是網域外後，用同樣的計算方式反推出 netmask 而已，網路上也有速成的公式可以套。因為這不是講網路架構的專書，所以有興趣的讀者建議自行參考相關書籍。

筆者這裡還要談一下虛擬 Private IP。這幾段 IP 是被保留的，因此直接透過這些 IP 是無法在網路上傳送資料的唷！這個時候就發明了 NAT 技術，透過 NAT 把內部虛擬 IP 偽裝到外部的真實 IP，這樣資料就可以傳送啦！且此時外部的真實 IP 無法直接連線到內部的虛擬 IP，這樣子還可以順便有防火牆的功能，也可以解決 IP 不足的問題。

A Class private IP : 10.0.0.0 ~ 10.255.255.255	預設 netmask 255.0.0.0
B Class private IP : 172.16.0.0 ~ 172.16.255.255	預設 netmask 255.255.0.0
C Class private IP : 192.168.0.0 ~ 192.168.255.255	預設 netmask 255.255.255.0

我們常用的 IP 分享器，無線網路分享器和防火牆等，其實是讓我們透過虛擬 IP 上網哩！通常我們使用這些機器都會順便啟用 DHCP 的功能，這樣連 user 端網路組態都不用自己設定。

DNS 概念

最後筆者以 DNS 最為結束。我們常常在上網的時候，都是打 www.ubuntu.org.tw 而不會去打它的 IP，這是因為網路上有一種 DNS server 幫我們把網址轉成 IP。所以 DNS server 一定要用 IP 的方式設定，不然的話就會有雞生蛋，蛋生雞的矛盾喔！

讀者可以回憶一下什麼是 IP、router、netmask 和 gateway 之間的關係，等一下設定馬上就會用到喔！

6.1.2 使用圖型介面來設定有線網路

若讀者你的網路系統有使用 IP 分享器，那大概就會開 DHCP 自動取得網路組態的功能。若讀者是學校單位或者商用單位，有配發固定的 IP，那也很適合用圖形介面來設定網路。但是目前 Ubuntu 沒有支援 PPPoE (PPP over Ethernet) 的預設圖形介面設定，除非你另外裝第三方軟體，不過筆者會建議你直接用文字介面設定 ADSL 撥接也很容易！文字介面的 PPPoE 撥接方式，會在 6.1.4 介紹到。筆者建議初學者，乾脆買一個便宜的 IP 分享器，這樣就有 DHCP 的功能，省時省力多啦！若要架站再用 DMZ 等方式設定。

要設定網路非常簡單，只要進入桌面選單的**系統→管理→網路**，然後填入你剛安裝時所輸入的個人帳號密碼，就可以看到如圖 6.8 的網路 GUI 設定介面。

使用 DHCP 來取得網路組態

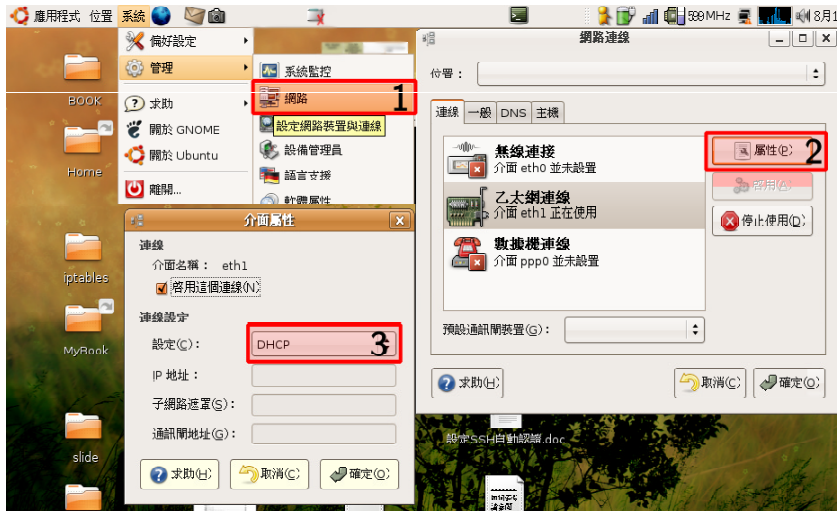


圖 6.8

若你有 IP 分享器或者是用 Cable modem 上網，只要如上圖步驟 3 的地方選擇為 DHCP 即可，通常連 DNS 等設定都可以自動從 DHCP 取得喔！所以筆者才建議大家買一個 IP 分享器是很實用的！

設定固定 IP 網路組態

若你使用固定 IP，那麼在上圖步驟 3 的地方需要選擇為固定 IP 地址即可。這時候把網管給你的 IP 地址，子網路遮罩（netmask）和閘道器地址（gateway）填上去。由於你是用固定 IP，要記得還要自己手動設定 DNS 呢！如圖 2，把網管告訴你的 DNS 填上去吧！若是不知道的話，填上 168.95.192.1 或 168.95.1.1 也可以用唷！建議填上兩組，這樣萬一在其中一台出問題後，還能幫你查詢 domain name 和 IP 的轉換哩！



圖 6.9

以上設定好後，就應該能正常上網了。這裡筆者再和讀者講一些進階的網路設定技巧，若你沒有把握的話，這些設定最好是不動啦！以免本來可以上網，卻被弄到連網路都出現問題。



圖 6.10

如圖 6.10，這一個設定對應到文字介面的 hosts，這些別稱的網址可以不透過 DNS 查詢，直接對應到 IP 地址。由於 IPv6 是下一代的網路 IP 技術，現在還沒有開始使用，所以讀者看到有 ip6 的別稱欄位，都可以砍了！這些欄位的 IP 地址長的和我們一般用數字的不一樣，讀者可別砍錯了！另外地址是 127 開頭的，是對應到本地主機，可別不小心砍到了。

在一般設定內，可以設定主機名稱和網域名稱，而 DNS 設定內還有搜尋網域可以設定，這通常在架站的時候會比較容易用到，筆者會在文字介面設定的時候詳細解說。

表 6-3 圖形介面網路設定和相對應的文字介面設定檔

GUI 設定位置	對應的文字介面設定檔
「連線」→「屬性」(IP、netmask 和 gateway)	/etc/network/interfaces
「一般」→「主機名稱」	/etc/hostname
「一般」→「網域名稱」	/etc/resolv.conf
「DNS」(DNS 伺服器和搜尋網域)	/etc/resolv.conf
「主機」(IP 地址和別稱對應)	/etc/hosts

6.1.3 使用圖型介面來設定無線網路

讀者需要先確定你的無線網路晶片已經驅動起來，並且如第二章討論的 firmware 也應該要裝起來了！如果你買那些 Ubuntu 有內建 firmware 的無線網路卡，就沒有這個煩惱啦！讀者可以透過指令 iwconfig 來確定是否正常驅動。

若你使用 Broadcom bcm43xx 系列的晶片，例如 Apple 的 ibook，雖然內建驅動程式，但是預設是缺少 firmware 的唷！你可以下載以下 firmware，然後安裝它吧！

http://www.cqis.ncku.edu.tw/~dbtsai/bcm43xx-firmware_1.1-0ubuntu1_all.deb

測試無線網路是否成功驅動：

```

dbtsai@ubuntu:~$ iwconfig
eth0      IEEE 802.11b/g  ESSID:"phys"  Nickname:"Broadcom 4306"
          Mode:Managed  Frequency=2.432 GHz  Access Point: 00:90:CC:B8:CF:2C
          Bit Rate:11 Mb/s   Tx-Power=10 dBm
          RTS thr:off   Fragment thr:off
          Link Quality=100/100  Signal level=2/3  Noise level=185/100
          Rx invalid nwid:0  Rx invalid crypt:583  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0
dbtsai@ubuntu:~$

```

若是你 `iwconfig` 沒有看到類似上面的訊息，那代表 Linux kernel 沒有抓到你的無線網路卡，這時候就真的麻煩了 XD。你這時候可以透過 `lspci` 來察看主機板內建的無線網路卡型號，或者透過 `lsusb` 來得知你的外接 USB 無線網卡型號。透過這些指令，你仔細找一下有關於 wireless 的訊息，上面應該有告訴你型號。再透過第二章選購無線網路硬體的技巧，來確認是否被 Linux 支援，若不支援的話，可能就要請你換一片了！

查詢你無線網路卡晶片：

```

dbtsai@ubuntu:~$ lspci
... (略) ...
0001:10:12.0 Network controller: Broadcom Corporation BCM4306 802.11b/g Wireless
LAN Controller (rev 03)
... (略) ...

```

若你的無線網路卡有被抓到，也成功驅動了，那麼現在測試一下是否可以抓到基地台！若是有查詢到基地台，你已經幾乎是成功啦！

查詢你可用的無線網路基地台：

```

dbtsai@ubuntu:~$ iwlist eth0 scanning
eth0      Scan completed :
          Cell 01 - Address: 00:90:CC:B8:CF:2C
                   ESSID:"phys"
                   Protocol:IEEE 802.11bg
                   Mode:Master
                   Channel:5

```



```
Encryption key:off
Bit Rates:54 Mb/s
Extra: Rates (Mb/s): 1 2 5.5 6 9 11 12 18 24 36 48 54
Quality=100/100 Signal level=-139 dBm
...(略)...
```

若是你 iwconfig 有抓到無線網路卡，但是怎樣都 scan 不到頻道的話，有可能是你的筆記型電腦為了省電，預設是把無線網路電源關閉的！這種筆記型電腦通常在機子身上會有無線網路電源指示燈，也有無線網路開關熱鍵，必須要按熱鍵才能打開電源，但這熱鍵在 Linux 上是沒有效果的，導致你無法掃描頻道。這時候可以進入主機的 BIOS 調整，讓無線網路電源一直開著，這樣就應該可以掃描到了！或者用 iwconfig wlan0 txpower on 來試試看！

接下來我們要開始安裝圖型介面的無線網路管理程式，現在先建議你把有線網路裝上吧！不然得要先在文字介面設定好無線網路。

✎ 安裝圖形無線網路管理介面（Gnome）：

```
sudo apt-get install network-manager-gnome
```

✎ 安裝圖形無線網路管理介面（KDE）：

```
sudo apt-get install knetworkmanager
```

安裝完後，登出再登入就會自動執行啦！或者你可以在指令列模式直接打 nm-applet，就會出現管理圖示哩！

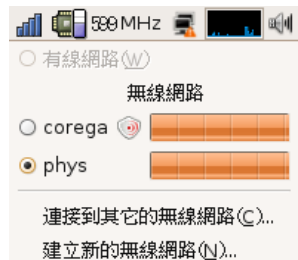


圖 6.11

如圖 6.11，它搜尋到兩個無線網路可以使用，依照你需要點選，就會自動連上去唷！有些無線網路是隱藏 ESSID 的，造成搜尋不到，你可以選擇**連接到其他的無線網路**，然後再輸入無線網路名稱和加密方式就可以連上啦！

一般咖啡店或者麥當勞等有無線網路的地方都沒有用加密，它們採用帳號認證的方式。若你家裡的基地台有加密的話，那搜尋到的名稱前面會有一個鎖頭圖示，點進去就會要你輸入密碼值。筆者建議用 WEP 就好，因為有些無線網卡的驅動程式還沒有穩定的實作 WPA 加密，根據筆者的測試，我連 WEP 幾乎都沒有問題，但是 WPA 有時候很不穩定！

要提醒讀者的是，使用 network manager 以後，建議 /etc/network/interfaces 就不要手動設定喔！否則會讓 network manager 失效哩！它還有一個特異功能，可以自動判斷你要用有線網路還是無線網路。例如你現在用無線網路，然後你再把有線的接上去，它會自動幫你轉到有線的並取得 IP 呢！所以筆者的筆記型電腦只裝 network manager，其他的設定我都不設定了，這樣很方便吶！

6.1.4 使用指令列模式設定 ADSL PPPoE 撥接

使用文字介面操作來連上 ADSL Modem 撥接其實很簡單，只需要幾個步驟就好啦！

 進入 PPPoE 設定畫面：

```
sudo pppoeconf
```

首先它會列出你可以使用的網路介面，選「是」即可。然後會如圖 6.12 掃描 ADSL 設備。

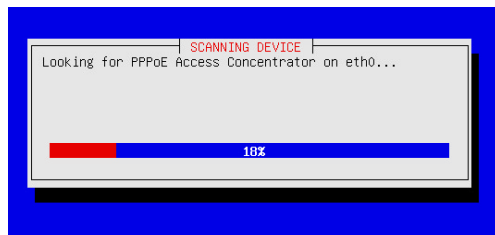


圖 6.12

接下來會如圖 6.13，它會確認你是否將設定檔備份，因為它等一下會修改到！
一般而言這個不用備份，所以選「是」即可！

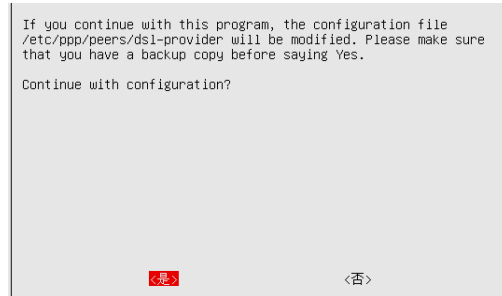


圖 6.13

如圖 6.14，這裡要輸入 ISP 給你的帳號。通常中華電信的帳號是 ID 號碼還要加上 @hinet.net，例如84500000@hinet.net，這樣才是你的帳號唷！這個各家 ISP 都有不同的作法。



圖 6.14

如圖 6.15，這裡輸入 ISP 給你的密碼。



圖 6.15

圖 6.16 提醒你，PPPoE 撥接後，一般會自動把 DNS 地址告訴你。這裡問你是否要把 DNS server 自動加到 resolv.conf。當然要選「是」，不然你還要額外設定 DNS 伺服器啲！

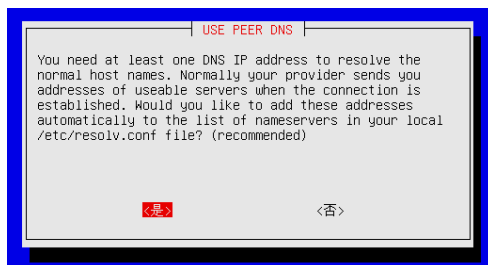


圖 6.16

圖 6.17 其實和我們先前網路概要所談的 MTU 有關。因為 PPPoE 協定是基於 PPP 的，所以會在網路層封包加上 PPP header 2byte，然後 PPPoE 自己的 header 有 6byte。而在傳輸層 TCP header 用了 20byte，網路層既有的 IP header 20byte，所以最後 PPPoE 可以拿來放資料的大小就只剩下 $1500-6-2-20-20=1452$ ，這個值就是所謂的 MSS 了！因此若你的 MSS 沒有設定到 1452 那麼大，有可能會因為封包向外傳遞時，會因為太大出不去，導致某一些網站無法進入。所以這個選項一定要選「是」啲！

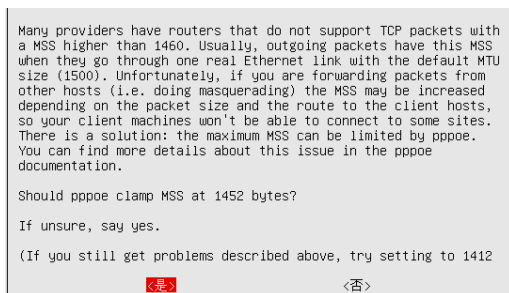


圖 6.17

圖 6.18 是問你要不要一開機就自動連線，讀者可以視你的需要決定啦！當然也是可以手動連線哩！

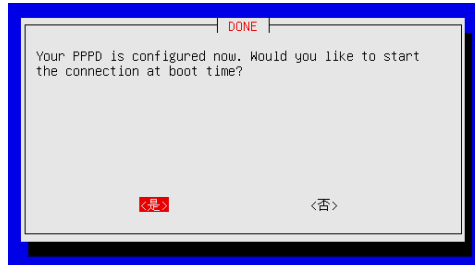


圖 6.18

現在應該已經可以上網啦！讀者可以試試看 `ifconfig`，會發現多了一個 `ppp` 裝置喔！

查詢 ppp 裝置：

```
dbtsai@ubuntu:/etc$ ifconfig
ppp0      Link encap:Point-to-Point Protocol
          inet addr:59.113.0.1  P-t-P:59.113.0.254  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP  MTU:1492  Metric:1
          RX packets:16632835  errors:0  dropped:0  overruns:0  frame:0
          TX packets:12857901  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:3
          RX bytes:1475487483 (1407.1 MB)  TX bytes:2359850091 (2250.5 MB)
          ... (略) ...
```

讀者可以透過以下指令來查詢 PPPoE 狀態。

指令：

```
plog
```

若你要手動把 ADSL 撥號斷線，可以透過這個指令。

指令：

```
sudo poff
```

若你要手動撥接 ADSL 上網，可以透過以下指令。

指令：

```
sudo pon dsl-provider
```

6.1.5 常用的網路相關指令

測試網路品質

當網路出現問題時，或者斷斷續續，我們常用 ping 來瞭解網路基本狀態。它所傳送的封包是 ICMP 封包，並要求對方主機給與回應，所以我們常常會用這些特性來得知網路狀態，故這個指令對網管而言，拿來 debug 網路錯誤是很好用的吶！

指令：

```
ping <參數> <地址>
# 地址可以是 IP 或者網址
# -b 對 broadcast IP 作 ping 的動作，會對該網段所有主機 ping
# -c 看你要對該地址連續 ping 幾次啦！若不加的話，會一直 ping 下去
# -i 每次 ping 的時間間隔，只有 root 才能小於 0.2 秒喔！
# -n 在 ping 的時候不做 IP 反查，這樣速度會比較快。
# -s 送出去的 ICMP 封包大小，預設為 56 byte，加上 8 byte 的 ICMP 表頭
# 就是 MSS 啦！若要看的是網路層大小，還要加上 20 byte 的 IP 表頭封包，
# 所以我們在測 MTU 的時候，要減去 28 才是 ICMP 封包大小喔。
# -M do 在 ICMP 封包上標記為 Don't Fragment，也就是在路由中不能拆開封包
# 常用在測量 MTU 之用。
# -M dont 剛好和上面不同，它代表著路由器可以把風包拆成比較小包的！
```

範例：

```
dbtsai@ubuntu:~ $ ping -c 2 168.95.192.1 # ping 中華電信 DNS 5 次
PING 168.95.192.1 (168.95.192.1) 56(84) bytes of data.
64 bytes from 168.95.192.1: icmp_seq=1 ttl=248 time=43.1 ms
64 bytes from 168.95.192.1: icmp_seq=2 ttl=248 time=42.0 ms

--- 168.95.192.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 41.920/42.663/43.385/0.587 ms

# 56(84) 那個代表了 ICMP 資料有 56 byte，84 是網路層大小，
# 是 56 + ICMP 8 + IP 20 = 84 byte，這個和 MTU 是算到同一層的單位喔！
# 64 byte 代表了 56 + ICMP 8，也就是扣除 IP 表頭的大小
# ttl 代表著封包經過幾個 node，每經過一個 node 後 TTL 會減 1，它的起始值
```

```
# 是從 255 開始，所以越接近 255 代表經過的 router 或 bridge 越少。
# time 43.1 ms 是回應時間，當然是越短越好啦！也代表你們之間的網路狀況很好。
```

當筆者網路出現問題的時候，會用以下步驟來除錯，讀者可以參考一下。

- ping 127.0.0.1：這是 ping 自己的 loopback 裝置，若連這個都不通的話，大概是整個 Linux 網路核心都有問題，這時候你大概要先檢查自己的電腦！
- ping <自己的 IP 地址>：上面的測試沒問題的話，這個通常也不會有什麼問題，除非網路卡有錯誤。
- ping <gateway>：看看區域網路對外的 router 有沒有通，搞不好是 router 這裡掛了！
- ping 168.95.192.1：這是中華電信的 DNS。雖然我測試的網路不一定是中華電信，但是它可靠性很高，常拿來當成我測試對象。若這裡不通的話，大概是網路對外了線路故障，如 ADSL 斷線等。若是 time 值很高，可能是區網內有人在抓 BT 或 P2P 導致網路不穩。

另外還有一個指令對於 debug 網路很有用，就是 tracepath 啦！常常我們有些網站可以連，但某些網站連不上去，這時候可以用這個方式來檢查到該台主機的路由有沒有問題。它的原理是用 ICMP 加上 ttl 去把經過的路由全部抓出來。但因為有些經過的路由會鎖 ICMP，所以有可能跑到一半會有沒有回應的路由器。

指令：

```
tracepath <地址>
# -n 在 tracepath 的時候不做 IP 反查，這樣速度會比較快
```

範例：

```
dbtsai@ubuntu:~ $ tracepath 168.95.192.1
1: 140.116.xx.xx (140.116.xx.xx) 0.195ms pmtu 1500
1: 140.116.xx.xx (140.116.xx.xx) 1.050ms
2: 140.116.xx.xx (140.116.xx.xx) 1.064ms
3: 140.116.xx.xx (140.116.xx.xx) 0.927ms
4: tn-cd-tanet-gw01.router.hinet.net (211.20.206.58) 1.555ms
5: tn-cd-c12r1.router.hinet.net (203.75.72.90) 1.028ms
6: tn-st-c12r11.router.hinet.net (220.128.26.34) 1.674ms
```

```

7:  ty-fo-c12r11.router.hinet.net (220.128.8.6)          asymm 10   7.307ms
8:  tp-s2-c12r11.router.hinet.net (220.128.1.30)           8.058ms
9:  tp-s2-c12r2.router.hinet.net (220.128.1.109)        8.465ms
10: tp-s2-c6r10.router.hinet.net (211.22.35.161)       7.719ms
11: 210.59.204.198 (210.59.204.198)           8.341ms
12: 210.59.204.210 (210.59.204.210)         7.215ms reached
Resume: pmtu 1500 hops 12 back 12

```

DNS 正反查詢

從網址查詢 IP，我們一般稱為 DNS 正解。若得知了 IP，想要查詢它的網址，一般稱為反解。而正解和反解可以是不一樣，因為我們可以申請很多個網域對應到同一個 IP，但是反解確要上層單位幫你設定！所以一般人很少有機會設定反解的，只有專業的網路用戶 ISP 才會幫你設定。所以有些檔垃圾信的方式是看反解網域，因為會發垃圾信的幾乎都是從那幾個反解網域出來的。

例如 Ubuntu 台灣站 www.ubuntu.org.tw 的正反解就不同，你可以從反解看到它放在台南縣網中心裡面。

指令：

```
nslookup <位址>          # 若是加 IP 的話就是反解，若加網址的話就正解
```

範例：

```

dbtsai@ubuntu:~ $ nslookup www.ubuntu.org.tw          # DNS 正解
Server:          168.95.192.1                          # 使用的 DNS 伺服器
Address:         168.95.192.1#53

Non-authoritative answer:
Name:   www.ubuntu.org.tw
Address: 203.68.102.240                                # 查詢出來的結果

dbtsai@ubuntu:~ $ nslookup 203.68.102.240            # DNS 反解
Server:          168.95.192.1
Address:         168.95.192.1#53

240.102.68.203.in-addr.arpa  name = www.opensource.org.tw. # 查詢出來的結果
Authoritative answers can be found from:

```



```
102.68.203.in-addr.arpa nameserver = mrtg.tnc.edu.tw.
102.68.203.in-addr.arpa nameserver = dns.tnc.edu.tw.
mrtg.tnc.edu.tw internet address = 163.26.200.2
dns.tnc.edu.tw internet address = 163.26.200.1
```

6.1.6 查詢主機目前的連線

若是你想要知道目前主機上有哪些 port 或服務在 listen，可以讓別人連進來，或主機上有哪一些連線，可以透過 netstat 來達成喔！

指令：

```
netstat
# -a 列出所有連線，包含 TCP, UDP 和 socket 通訊
# -l 列出是主機在 Listen 的連線，通常為伺服器軟體
# -n 在 netstat 的時候不做 IP 反查，這樣速度會比較快
# -p 列出使用該連線的軟體名稱和 PID
# -t 列出 TCP 通訊部份連線
# -u 列出 UDP 通訊部份連線
```

範例：

```
dbtsai@ubuntu:~ $ netstat -tlnl
Active Internet connections (only servers)

```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN	7840/mysqld
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN	9779/apache2
tcp	0	0	0.0.0.0:21	0.0.0.0:*	LISTEN	5372/vsftpd
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	5355/sshd

如以上範例，我們可以得知主機上有哪些 port 被打開在 Listen。例如這台是筆者的伺服器，所以可以看到幾個常用的 port 都被打開啦！這個用來 debug 你的伺服器很好用，也可以知道目前有多少連線到你的伺服器上。

查詢設定網路卡組態

筆者已經在很多個不同的場合介紹到 ifconfig 這個指令了，這裡正式的和讀者介紹一下！它的功能很多，包含可以查詢目前網路卡組態，或者設定一些東西，

其實我們把設定檔寫在 `/etc/network/interfaces` 裡面後，也是透過 `ifconfig` 去真正設定的啲！

指令：

```
ifconfig <interfaces> <options>
# 可用的 options 有 netmask, broadcast 和 mtu 等，也可以加 IP address
# 若加上 down 或 up 的話，就是啟用或關閉該網路裝置喔！
# 甚至可以用 ifconfig eth1 hw ether 11:22:33:44:55:66 來改網路卡 MAC 卡號
```

範例：

```
dbtsai@ubuntu:~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:11:24:29:3C:BD
          inet addr:192.168.0.141  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:112251 errors:0 dropped:72593 overruns:0 frame:0
          TX packets:111636 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:88025996 (83.9 MiB)  TX bytes:10474558 (9.9 MiB)
          Interrupt:52 Base address:0x8000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1352 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1352 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21566778 (20.5 MiB)  TX bytes:21566778 (20.5 MiB)
```

如範例，你看到了基本的網路設定資料了吧！筆者就簡單的介紹不同欄位的意思。

HWaddr	這就是網路卡 MAC address，基本上這是全世界獨一無二的，但是你可以透過 <code>ifconfig</code> 修改。例如本例就是 <code>00:11:24:29:3C:BD</code> 。
inet addr	在範例中的 <code>192.168.0.141</code> 就是 <code>eth0</code> 的 IP 喔！
RX packets	代表了網路卡啟用後接收到多少個封包
TX packets	代表了網路卡啟用後傳送出去多少個封包
MTU	從網路層算起來最大封包的大小

那個 `lo` 裝置就是所謂的 loopback 設備，它常常用在主機內部的 TCP/IP 程式溝通，就不需要經過網卡啦！所以它的 IP `127.0.0.1` 常用在主機自己連到自己的時候用，且因為直接在主機內傳遞，MTU 就會是 TCP/IP 的理論最大值，所以效率上來說會比連到 `eth0` 還好很多喔！

如以下範例，`ifconfig` 也可以拿來直接設定該裝置的 IP，但記得還要另外用 `route` 加上 `gateway` 喔！`resolv.conf` 的 DNS 組態也要另外設定。

設定固定 IP：

```
sudo ifconfig eth0 192.168.0.150 netmask 255.255.255.0 # 設定 IP 和 netmask
sudo route add default gw 192.168.0.1 # 設定 gateway 閘道器
```

檢視你的路由表

在網路概論的地方筆者提過，本機電腦是用 `netmask` 來決定封包是不是區網內，若不在區網內的話，就會送到 `router` (`gateway`) 去轉送到其他網域。前面教學中已經討論到如何設定 IP 和 `netmask`，那 `route table` 要在哪裡設呢？答案就在 `route` 這個指令。

指令：

```
route # -n 在顯示 route table 時不進行 IP 反解，速度會比較快
```

路由表範例：

```
dbtsai@ubuntu:~$ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.0.0      0.0.0.0         255.255.255.0  U        0      0      0 eth0
0.0.0.0          192.168.0.1    0.0.0.0        UG       0      0      0 eth0
```

如範例，這是筆者在 IP 分享器後面所用的路由表。第一個 `Destination` 代表了目的地網段的代號。例如第一條是 `192.168.0.0`，代表了區域網路網段的開始，它不需要經過 `gateway` 就可以直接從 `switch` 連到，所以它的 `gateway` 值是 `0.0.0.0`。第一條的 `genmask` 其實就是 `netmask`，代表了該網段的大小。

第二條規則的 `Destination` 是 `0.0.0.0` 且 `genmask` 是 `0.0.0.0`。但任何 IP 和 `0.0.0.0` 這個 `netmask` 做 AND 邏輯運算，永遠都是 `Destination` 的網段 `0.0.0.0`，

所以它代表了世界上所有網路 IP 的集合。這意味著這條規則告訴電腦，不管目的地在哪裡，都要往 gateway 送。

讀者或許奇怪，第二條規則怪怪的啊！不是說區網只要直接連線，不需要經過 gateway 嗎？其實是這樣的，當封包在判別路由時，會先從第一條依序往下判別，當成立的時候，就會送出去不再判別。所以在這個例子中，送往區網的封包在第一條規則就被判別出來了，當然就用不到第二條啦！若沒有第一條怎麼辦？這時候電腦會把不管區網內或區網外的封包都送到 router，然後 router 會幫我們把目的地是在區網內的封包再送回來，不過這樣一來一往，網路效能就不好了！筆者就遇過朋友把自己網段的 netmask 設成 255.255.255.255，同學們都在區網內抓好康的，但他雖然連得到，卻速度特別慢，聰明的讀者，你知道原因了嗎？

若你不是用設定檔的方式或 GUI 介面設定網路，那 route table 就要自己設了，這裡不考慮大型網路環境有好幾個不同的路由，要依照不同目的地網域選擇路由的情形，只考慮一般環境會用到的設定。

刪除路由範例：

```
dbtsai@ubuntu:~ $ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.0.0      0.0.0.0         255.255.255.0   U      0      0      0 eth0
0.0.0.0          192.168.0.1    0.0.0.0         UG     0      0      0 eth0
dbtsai@ubuntu:~ $ sudo route del -net 0.0.0.0 gw 192.168.0.1
# 0.0.0.0 可以用 default 替代！
dbtsai@ubuntu:~ $ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.0.0      0.0.0.0         255.255.255.0   U      0      0      0 eth0
```

如以上範例，我們可以把 0.0.0.0（可用 default 替代）那條路由刪除。當網路不通時，先檢查路由設定有沒有錯誤，若有錯的話，可以用以上方式把它砍了！再用接下來的方式重新設定路由吧！

新增路由範例：

```

dbtsai@ubuntu:~ $ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.0.0      0.0.0.0         255.255.255.0   U      0      0      0 eth0
dbtsai@ubuntu:~ $ sudo route add -net default gw 192.168.0.1
# 最後可以加 dev eth0 指定用這張網路卡。
# 當公司內部有另外一個小網域時，可以用
# add -net <目標網域> <netmask> gw <gateway> 來指定該網域的 route table，
# 最後再加上 default 的路由。
# 若是要指定單部主機路由的話，可以用 add -host IP <netmask> gw <gateway>
dbtsai@ubuntu:~ $ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.0.0      0.0.0.0         255.255.255.0   U      0      0      0 eth0
0.0.0.0          192.168.0.1     0.0.0.0         UG     0      0      0 eth0

```

使用 DHCP 取得你的網路組態

當臨時到一個使用 DHCP 的地方，你也不想改動網路設定檔，那有個簡便的方式讓你自動取得 IP 等網路組態。

DHCP 指令：

```
sudo dhclient eth0 # 在網卡 eth0 上取得 IP, netmask, gateway 和 DNS 設定
```

通常使用 DHCP 的網路，都會自動提供所有設定的訊息，所以一但成功取得 IP，其他的你什麼都不需要設定了。若你要每次都用 DHCP，在前面 GUI 或者下一小節都有介紹。

6.1.7 使用傳統設定檔的方式設定有線網路

表 6-4 重要的網路設定檔

設定檔	說明
/etc/network/interfaces	設定網路 IP、Netmask 和 Gateway 或使用 DHCP 等網路組態
/etc/resolv.conf	設定 DNS 伺服器和本機的網域名稱或網域查詢列表
/etc/hostname	主機名稱設定 (hostname)
/etc/hosts	建立靜態主機名稱與 IP 對應

表 6-4 列出了各個重要的網路設定檔，筆者將會逐一的解釋其功能與用途。一般沒有特殊需求的話，通常設定了 interfaces 和 resolv.conf 就能讓網路正確的運作！若拿來架站或者架 NAT 的話，hostname 和 hosts 就很重要了，錯誤的設定往往會導致網路效能很差。

設定檔 interfaces

首先筆者的主機有兩片網路卡 eth0 和 eth1。假設第一片要設定成固定 IP，第二片要設成使用 DHCP 來取得 IP 組態。這個範例將會示範給讀者了解如何依照我們所描述的例子來設定網路。讀者要設定固定 IP 的時候，向你們的網管取得網路 IP 吧！不要亂設定喔！我們系上就曾經有個很天兵的教授把他的主機設成 140.116.xxx.1，那台剛好和我們系上的 Web server 一樣 IP，害我們網頁主機好幾天打不開！最後找到兇手了，那位教授還說書本上的範例就是用 xxx.xxx.xxx.1 阿！為什麼他不能用？真是有夠 Orz 的！

現在用你最熟悉的文字編輯器把設定檔打開吧！

 編輯 interfaces 設定檔：

```
sudo gedit /etc/network/interfaces
```

 設定檔 interfaces 內容範例：

```
auto lo # 啟用 loopback 設備。注意，請不要改動它！
iface lo inet loopback # 設定 loopback 設備。注意，也不要改動到它！
```

```

auto eth0                # 啟用 eth0 網路卡，若你有其他網路卡，可以直接
auto eth1                # 加在後面變成 auto eth0 eth1 或者獨立成一行設定

iface eth0 inet static   # 設定 eth0 這張網路卡為固定 ip
    address 192.168.1.5   # 設定 IP 地址
    netmask 255.255.255.0 # 設定子網路遮罩
    gateway 192.168.1.1  # 設開道器
    network 192.168.1.0  # 設定網路代號，不一定要設定，可有可無！
    broadcast 192.168.1.255 # 設定廣播地址，不一定要設定，可有可無！

iface eth1 inet dhcp     # 設定 eth1 這張網路卡從 DHCP server 取得 IP

```

依照上面範例，讀者應該覺得很簡單吧！記得喔，lo 是內部回圈用的虛擬網路設備，不要去修改到它喔！還有在設定固定 IP 時，network 和 broadcast 其實是可以從 IP address 和 netmask 推算出來，所以這兩個值其實是可以不用設定的，這裡列出來是為了增加說明的完整性。

這裡設定好了以後，需要重新啟動網路介面，這樣才能讓新的網路組態生效。很多讀者會覺得奇怪怎麼改完了以後網路設定沒有更變吶！甚至有人因此重新開機來使網路設定生效。記得啦，Linux 是不需要如此常重新開機，只要執行以下指令就可以讓新的設定生效啦！

重新啟用網路介面：

```
sudo /etc/init.d/networking restart
```

也可以只重新啟動單一網路設備：

```
sudo ifdown eth0 # 關閉 eth0 這個裝置
sudo ifup eth0 # 啟用 eth0 並從 interfaces 讀取設定檔
```

設定檔 resolv.conf

筆者有遇過設定好了網路，明明 ping IP 會通，但打網址就不會通！這是因為忘了設定 DNS 伺服器了，導致網址不能透過 DNS server 轉成 IP。

 編輯 resolv.conf 設定檔：

```
sudo gedit /etc/resolv.conf
```

 設定檔 resolv.conf 內容範例：

```
nameserver 168.95.192.1 # DNS 第一台主機  
nameserver 168.95.1.1 # DNS 第二台主機
```

通常我們習慣設定兩台 DNS 主機，避免萬一哪一台故障了，導致你的網路出現不正常。筆者習慣是第一台設定自己單位架的 DNS，第二台可以設定中華電信的 168.95.192.1，尤其是中華電信的 DNS 對不管哪裡來的 ISP 使用者都提供服務，並且查詢很快。在設定檔中，最多只能設到三台。它的查詢方式是，會先向第一台詢問，如果第一台沒有回應，則自動向第二台查詢。

設定檔中還有其他可以設定的選項，如 search 和 domain，它們的功能其實很接近。注意喔！它們兩個只能設定一個，也就是這兩個選項是互斥的。若同時設定在 resolv.conf 裡面，就以最後一個有效喔！

 設定檔 resolv.conf 中使用 search 範例：

```
search bd-dvd.org emook.org  
nameserver 168.95.192.1 # DNS 第一台主機  
nameserver 168.95.1.1 # DNS 第二台主機
```

如範例，筆者在 search 中設定了 2 個網域，此時筆者 ping www 或者連線到 www 的時候會發生什麼事呢？首先它會嘗試用 DNS 正解 www.bd-dvd.org，若它存在的話，就直接送回它的 IP 地址。若是不存在的話，再嘗試 www.emook.org。也就是它會對你要正解的網址先加上 search 內的網域查詢，若沒有的話才會直接用你給它的網址查詢。這個技巧在連線到同網域的主機常常用到，我們常把自己區網的網域設在 search 裡面，這樣子就只要打主機名稱就可以連上啦！

這有個缺點，若你現在要連 www.gmail.com，它首先會用 DNS 正解 www.gmail.com.bd-dvd-.org，若是沒有的話在依序查詢下去，最後才會查詢到 www.gmail.com。沒錯，就是那麼笨！所以若你 search 裡面設定的網域不是區網內的話，且你沒有自己的 DNS server，這樣子這些查詢都會到外面的 DNS 查詢，並會因為有很多需要 search 的網域而導致查詢很多次，如此網路上會都是 DNS 封包而讓網路效能降低。所以你大概也不會想放太多網域在 search 裡面。

當有兩個以上的網域在同一個區網時，設定檔裡面前後順序就很重要了，因為它會先 try 前面那一個網域，所以建議先放主機本身的網域，這樣子才不會搞混。

設定檔 resolv.conf 中使用 domain 範例：

```
domain emook.org
nameserver 168.95.192.1      # DNS 第一台主機
nameserver 168.95.1.1      # DNS 第二台主機
```

使用 domain 的話幾乎和 search 一樣，不過它只能設定一個，也就是主機本身的網域，不能再加上其他的。它查詢的規則是和 search 一樣，會把你查詢的網址加上 domain 裡面設定的網域，最後才會查詢到你的網址。通常筆者是習慣設定 search 而不設定 domain。

設定檔 hostname 與 hosts

所謂的 hostname 是我們幫主機取的名子。例如我有一台主機的網址為 ubuntu.emook.org，那麼 ubuntu 是這個主機的 hostname，而 emook.org 是這個主機的網域。因此 hostname 是一定沒有 .(dot) 符號的，有些人會把 /etc/hostname 設成完整的網址名稱，這是不對的唷！

/etc/hostname 其實和 /etc/hosts 有很大的關連性，因為我們在 hostname 只設定了主機名稱，但沒有帶有網域名稱，所以若要查主機的完整網址 Fully Qualified Domain Name (FQDN)，就必須要 and hosts 配合。而 FQDN 的表達方式為《主機名.網域名》，所以 ubuntu.emook.org 就是 FQDN 表示法。

要設定 hostname 很簡單，直接把 hostname 寫到設定檔即可。

編輯 hostname 設定檔：

```
sudo gedit /etc/hostname
```

設定檔 hostname 範例：

```
ubuntu
```

由於 hostname 這個設定檔是開機時，在跑 /etc/rcS.d/etc/rcS.d/S02hostname.sh 的時候把它從 /etc/hostname 載入到系統。所以當修改過設定檔後不會馬上生效，除非使用以下指令更新。

更新 hostname 到系統：

```
sudo hostname -F /etc/hostname
```

你可以如以下範例查詢目前的 hostname。

查詢系統的 hostname：

```
dbtsai@ubuntu:~$ hostname
ubuntu
```

當查詢完整的 FQDN 時，出現以下錯誤，通常是因為 hosts 或 resolv.conf 沒設定好！

查詢系統的 FQDN：

```
dbtsai@ubuntu:~$ hostname -f
hostname: Unknown host
dbtsai@ubuntu:~$
```

這對於主機架設很重要，系統很多重要的服務如 Apache 或 MySQL 都需要有完整的 FQDN 存在，否則會出現錯誤。當你用 hostname -f 去查詢系統的 FQDN 時，系統會用你的 hostname 設定的名稱去和 hosts 比對來找出網域名稱形成完整的 FQND。若是在 hosts 裡面比對不出來的話，系統會再找 resolv.conf 裡的 search 或 domain 來當成網域名稱。

所以我們現在來看看 /etc/hosts 長得怎麼樣。

編輯 hosts 設定檔：

```
sudo gedit /etc/hosts
```

設定檔 hosts 範例：

```
127.0.0.1    localhost.localdomain  localhost
192.168.1.10  ubuntu.emook.org      ubuntu
```

```
192.168.1.20    node1
192.168.1.21    node2
192.168.1.22    node3
```

如 hosts 範例檔，筆者第二行是本機伺服器的設定，第二欄 ubuntu.emook.org 是主機的完整名稱，最後一欄 ubuntu 是主機名或別名。

後面 node1 等是為了讓區網內虛擬 IP 連到主機的時候可以反解，所謂的反解是從 IP 查詢主機名稱。當伺服器架在虛擬 IP 內，而虛擬 IP 網段內的其他電腦要連到伺服器時，伺服器會先嘗試反解，若 hosts 裡面沒有把這些 client 端的反解寫上去，會造成伺服器嘗試反解但解不出來，這段時間會耗時許久！因此當伺服器和 client 端都在虛擬 IP 網段時，建議把所有 client 的 IP 加到伺服器的 hosts 上吧！

成功的查詢系統 FQDN：

```
dbtsai@ubuntu:~$ hostname -f
ubuntu.emook.org
dbtsai@ubuntu:~$ hostname
ubuntu
```

如以上範例，我們建立了正確的 hostname 和 hosts 以後，可以查詢到系統的 FQDN 而不會有錯誤。若是讀者怎樣弄都匹配不起來，可以把網域寫在 resolv.conf 的 search 或 domain，它在最後真的無法匹配時，會用 search 或 domain 的網域加到你的 hostname 來形成 FQDN，這樣總比有錯誤好！在伺服器中，建議要讓 hostname 和 hostname -f 都能正常顯示，這樣才不會讓某些 server 服務出錯。

6.1.8 使用傳統設定檔的方式設定無線網路

我們來先了解幾個無線網路常用的指令。筆者要提醒你的是，若是你把無線網路設定寫死在 interfaces 內，那麼 network-manager 無線網路管理程式就會失效囉！筆者還是建議你無線網路這種東西就用 GUI 就好啦，除非有其他的特殊需求。

第一個指令在無線網路 GUI 那裡提過了，你可以用以下指令來掃描可以使用的無線網路。

掃描無線網路頻道：

```
sudo iwlist eth0 scanning
```

設定無線網路最重要的指令是 `iwconfig`，讀者可以透過它來了解目前無線網卡的狀態，若是你要加入一個無線網路，可以用以下指令！

加入無線網路頻道：

```
sudo ifconfig eth0 up # 啟用 eth0 這張無線網卡
sudo iwconfig eth0 essid phys key 123456789
# phys 是筆者 AP 的 essid，當然你要換成你所掃描到的
# 後面的 key 就是 WEP 密碼啦！
```

連上了以後，你可以透過 DHCP 來取得 IP 地址，如以下指令。

自動取得 IP：

```
sudo dhclient eth0
```

通常無線網路的環境都是使用 DHCP。若是你堅持的話，可以不要用 DHCP，使用以下指令來使用固定 IP，不過此時你還要自己設動 `resolv.conf` 來讓 DNS 可以正常運作唷！

設定固定 IP：

```
sudo ifconfig eth0 192.168.0.150 netmask 255.255.255.0 # 設定 IP 和 netmask
sudo route add default gw 192.168.0.1 # 設定 gateway 閘道器
```

每次都這樣設定是很麻煩的，所以這些設定可以直接寫到 `/etc/network/interfaces` 裡面！

設定無線網路組態到 `interfaces`：

```
auto eth0 # 啟用無線網路設備
iface eth0 inet static # 設定為固定 IP，若要 DHCP 只要把 static 改成 dhcp
wireless-essid phys # 設定無線網路名稱
wireless-key 123456789 # 設定無線網路密碼
```

```
address 192.168.0.150      # 若為 DHCP，以下都不需要設定！
netmask 255.255.255.0
gateway 192.168.0.1
```

6.1.9 最佳化網路卡 MTU 與關掉 IPv6

網路概論的地方有談過 MTU 是從網路層開始算的最大封包大小。在乙太網路的架構中，傳送 TCP 封包最多可以有 1460byte 的應用層資料，但每個封包在實體網路需要用 1524byte 來傳遞！也就是有大概 4% 的頻寬是浪費在表頭身上。

在區域網路傳送封包而言，當然是 MTU 越大越好，它節省的不只是那 4% 表頭大小，因為當 MTU 大時，需要切割的封包就變少，如此還可以節省 CPU 處理封包重組的時間。若你想要把 MTU 調大，你要先確定你的區域網路都是用 giga 網路卡，且 switch 有支援 Jumbo frame，而且網段內最好不要有 10/100 的 ethernet 網卡。故這種最佳化技巧其實比較適合用在實驗室，或者平行運算 cluster 主機。在筆者的測試，當把 MTU 開到 9000 時，除了省下 3% 的表頭外，更因為封包減少的關係，網路平均約有 10% 效能的改善！但並不是每一塊 giga 網路卡 MTU 都可以上去 9000，例如筆者桌上型內建的 giga 網卡 MTU 最高只能設到 7000。所以最大 MTU 的值，讀者可以慢慢調上去，順便測試效能。為了避免網路出錯，建議網段上所有電腦的 MTU 都設成一樣唷！

備註 某一些 giga 網卡 MTU 可以上到 16000，讀者可以自己試試看！

修改網卡 MTU：

```
dbtsai@ubuntu:~$ sudo ifconfig eth0 mtu 9000 # 修改 eth0 網卡的 MTU 為 9000
dbtsai@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:11:24:29:3C:BD
          inet addr:192.168.0.141  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:9000  Metric:1
          ... (略) ...
```

修改後，讀者可以透過 ifconfig 看 MTU 那個欄位，應該已經變成了 9000 了吧！

更改後，讀者當然會想要試試看實際上封包傳輸時最大 MTU 可以到多少，讀者可以透過以下指令來測試區網內傳遞的 MTU 唷！

察看實際傳輸時的 MTU :

```
ping -c 1 -s <payload byte> -M do <IP 地址>
# -M do 會在 ICMP 封包標上不要 fragmentation，這樣路由器就不會幫你切割封包
# -c 的話告訴 ping 只要測試一次唷！
# -s 參數為 ICMP payload 大小，他的值為 MTU 減掉 ICMP 表頭大小 28 唷！
# 若你的 MTU 大小設為 1500，那個 payload byte 就是 1472
```

若 switch 和對方主機都能以 MTU 9000 工作的話，訊息如下範例。注意，給的 payload 值一定要扣掉 28 喔！

範例：

```
dbtsai@ubuntu:~$ ping -c 1 -s 8972 -M do 192.168.0.30
PING 192.168.0.30 (192.168.0.30) 8972(9000) bytes of data. # 8972(9000) 為MSS(MTU)
8980 bytes from 192.168.0.30 : icmp_seq=1 ttl=250 time=1.12 ms
...(略) ...
```

若區網中 switch 或者對方不支援的話，會有以下範例的錯誤訊息，此時讀者可以查詢 switch 是否支援，或者把 MTU 調小一點看看。

範例：

```
dbtsai@ubuntu:~$ ping -c 1 -s 8972 -M do 192.168.0.30
PING 192.168.0.30 (192.168.0.30) 8972(9000) bytes of data. # 8972(9000) 為MSS(MTU)
From 192.168.0.141 icmp_seq=1 Frag needed and DF set (mtu = 9000)
...(略) ...
```

每次都要設定 MTU 很麻煩，所以讀者也可以把 MTU 的值寫到 /etc/network/interfaces 中，這樣開機就會自動設定 MTU。

在定檔 interfaces 內寫入 MTU 值：

```
iface eth0 inet static # 設定 eth0 這張網路卡為固定 ip
pre-up /sbin/ifconfig eth0 mtu 9000 # 設定 eth0 MTU
    address 192.168.1.5 # 設定 IP 地址
    netmask 255.255.255.0 # 設定子網路遮罩
    gateway 192.168.1.1 # 設開道器

iface eth1 inet dhcp # 設定 eth1 從 DHCP 取得 IP
pre-up /sbin/ifconfig eth1 mtu 9000 # 設定 eth1 MTU
```

畢竟大部分的人都不會在家裡用 giga 網卡和 switch 吧！筆者這裡也講一下 MTU 對於 ADSL 的影響。由於 ADSL 使用 PPPoE 協定，而 PPPoE 的表頭又會多佔了 8 byte，因此在乙太網路架構中，MTU 變成了 1492。若讀者使用 ADSL 撥接的話，ifconfig 就會看到 ppp 設備的 MTU 為 1492。

封包切割不是什麼好事情，因為 router 把封包切割了以後，若其中一個小封包錯誤，就需要整包重傳。若 MTU 大小是 1500，當經過 MTU 1492 的 ADSL 線路，封包就會變成兩個，一個大小是 1492byte，另外一個大小 48byte，第二個封包實際上是只有 8byte 的內容，其餘的 40byte 都是表頭！這是多麼沒有效率的事情啊，一但封包經過切割，除非到目的地是不會重組回去的！於是就有了 MTU 路徑探索 (Path MTU Discovery) 技巧來得知本機到對方電腦可用的最小 MTU，這樣就避免了封包切割的問題。但是在路程中有些 router 不支援，或者你對外的 router 設計不良，常常就會產生路徑探索錯誤。有些人用 ADSL 時，NAT 內電腦會連不上某些網站，或者無法傳輸檔案，此時就可以手動把你連到 router 的那張網路卡 MTU 改成 1492 就可以連了。

最後，IPv6 是下一代的 IP 技術，目前還在實驗中，但是 Ubuntu 預設已經有開啟了，為了減少系統資源，建議把它關掉吧！讀者可以直接在 /etc/modprobe.d/blacklist 裡面加上 blacklist ipv6 即可。

6.2 安裝符合你平台的核心與顯示卡－ OpenGL 驅動設定

假如讀者是安裝 PC 版本，也就是 32 位元版的 Ubuntu，預設是使用 386 的核心，但各位讀者的 CPU 不可能是那麼舊的，所以筆者會先教大家如何升級到比較符合你平台的核心，來獲得更好的效能。


顯示卡驅動方面，筆者寫到這裡時，Intel 將 965 晶片組的 3D 繪圖卡的驅動程式開放所有原始碼。目前已經整合到 Xorg 的程式碼開發中。這裡筆者要對 Intel 給與高度的讚揚和肯定（Intel 也僱用若干 Xorg 核心開發者）。

雖然 AMD (ATI) 和 Nvidia 有提供 Linux 的驅動程式，但是沒有提供原始碼。因為 AMD (ATI) 和 Nvidia 沒有提供原始碼，所以無法直接整合到 Linux 系統裡面，這導致常常會在升級的時候出現核心不相容等情形，並且他們使用的技巧也

游走在 GPL 授權邊緣。且他們只提供主流平台的驅動程式，如筆者使用 PowerPC 他們就沒有提供，讓筆者的電腦幾乎沒有 3D 加速的能力。這些問題都會在驅動程式開放原始碼解決，期許 Intel 的開放策略能讓 AMD (ATi) 和 Nvidia 能夠跟進，如此一來會讓 Linux 在桌面上應用更加寬廣。


6.2.1 選擇適合你平台的核心

讀者可以透過以下指令得知你所使用的 kernel 核心版本。

 得知目前使用的核心平台：

```
dbtsai@ubuntu:~$ uname -r # 如範例，筆者目前用的是 386 版預設的核心
2.6.17-11-368
```

我們可以透過以下指令來得知可以用的核心平台有哪幾種。

 得知所使用的核心平台：

```
dbtsai@ubuntu:~$ apt-cache search linux-image
linux-image-386 - Linux kernel image on 386. # 虛擬套件
linux-image-686 - Linux kernel image on Ppro/Celeron/PII/PIII/PIV. # 虛擬套件
linux-image-k7 - Linux kernel image on AMD K7. # 虛擬套件
linux-image-server - Linux kernel image on Server Equipment. # 虛擬套件
linux-image-2.6.15-14-386 - Linux kernel image for version 2.6.15 on 386 . # 實體核心
linux-image-2.6.15-19-386 - Linux kernel image for version 2.6.15 on 386. # 實體核心
... (略) ...
```

如讀者所看到的，那些 linux-image-386,686,k7 是虛擬套件，幫我們相依到最新版的核心。而後面那些有版本號的是真正的核心套件，請不要直接安裝他們唷，不然不會自動升級核心哩！

例如筆者這台電腦使用 AMD K8 CPU，但是因為某些相容性（如 Flash player, win32codec 等）的問題，我選擇使用 32 位元 Ubuntu，那麼我就需要用 K7 那個核心來最佳化 kernel 效能！

 安裝新的核心：

```
dbtsai@ubuntu:~$ sudo apt-get install linux-image-k7
```


若讀者是用 Intel P4 系列，那可以安裝 linux-image-686。或要使用於 server 用途，那麼裝 linux-image-server 會對 server 使用環境最佳化。

若讀者是使用 64 位元版本或 PowerPC 版本的 Ubuntu，那麼你也可以用以上指令來列出可用核心，例如在 64 位元版上有 linux-image-amd64-k8 這一類可以裝的核心。

6.2.2 Nvidia 顯示卡驅動程式安裝

Nvidia binary 驅動程式已經內建 Ubuntu 裡頭，不需要在自己上網下載以及進行編譯。


首先要先裝驅動程式。

 安裝 nvidia 驅動程式：

```
sudo nvidia-glx
```

第二步是安裝 nvidia 的核心模組，這和你所使用的核心有關，也就是上一小節所提及的核心平台版本。

你可以透過以下指令來查詢可以使用的核心模組。但注意，這和裝核心是一樣的，有分實體套件和虛擬套件，若你裝到實體套件的話，而核心有裝虛擬套件，那麼更新的時候有可能核心會升級但是 nvidia 所用的核心模組沒有升級，導致你的 Xwindow 開不起來！

 得知所使用的核心平台：

```
dbtsai@ubuntu:~$ apt-cache search linux-restricted-modules
linux-restricted-modules-2.6.15-26-686 - ..... # 實體模組套件
linux-restricted-modules-2.6.15-26-k7 - ..... # 實體模組套件
linux-restricted-modules-386 - Restricted Linux modules on 386. # 虛擬套件
linux-restricted-modules-686 - Restricted Linux modules on PII/PIII/PIV. # 虛擬套件
linux-restricted-modules-k7 - Restricted Linux modules on AMD K7. # 虛擬套件
...(略)...
```

所以讀者要選擇一個虛擬套件，且和核心相同平台的裝上去。你這裡也可以順便確認一下，核心是否也有裝到虛擬套件，重點是兩個都需要是裝虛擬套件（metapackage）否則很容易因版本不一致而出錯！

以筆者目前使用的 K7 平台為例，需要使用以下指令來裝模組。

安裝核心模組：

```
sudo apt-get install linux-restricted-modules-k7
```

接下來要讓驅動程式啟用，這有三種方式，推薦你使用第一個方式，若出現錯誤的話，再使用第二種或第三種方式吧！其實以下三種方式都是修改 /etc/X11/xorg.conf，把裡面 nv 那一個值改成 nvidia 而已！

方式一：

啟用驅動程式：

```
sudo nvidia-glx-config enable
```

若讀者自己手動修改過 /etc/X11/xorg.conf 的話，它會出現錯誤訊息，這時候可以修正 md5sum 或者手動設定。出錯的原因是因為你修改過 xorg.conf 這個檔案，導致它的 md5 hash 值和系統原先的不一樣。

錯誤訊息：

```
Error: your X configuration has been altered.
This script cannot proceed automatically. If you believe that this
not correct, you can update the md5sum entry executing the following
command:
md5sum /etc/X11/xorg.conf | sudo tee /var/lib/x11/xorg.conf.md5sum
otherwise edit manually /etc/X11/xorg.conf to change the Driver section
from nv to nvidia.
```

若要採用修正 md5sum 並且啟用驅動程式的方式，只要使用以下指令即可。

修正錯誤並啟用驅動程式：

```
sudo md5sum /etc/X11/xorg.conf | sudo tee /var/lib/x11/xorg.conf.md5sum
sudo nvidia-glx-config enable
```

方式二：

若是想要手動啟用的話，讀者可以編輯 `/etc/X11/xorg.conf` 這個設定檔，它幾乎掌管了所有 X server 的設定。

設定檔 `xorg.conf` 內容：

```

... (略) ...
Section "Device"
    Identifier      "NVIDIA Corporation NV34 [GeForce FX 5200]"
    Driver          "nv"      # 把這裡個 nv 改成 nvidia 吧！
    BusID          "PCI:1:0:0"
EndSection
... (略) ...

```

讀者找到了 Driver 裡面設定值 `nv` 後，把它改成 `nvidia`，存檔就好啦！

方式三：

若讀者懶得修改 `xorg.conf` 的內容，可以透過重設 X server 時，可以讓你選擇硬體驅動型號，這時候選擇 `nvidia` 就好啦！

重設 X server：

```
sudo dpkg-reconfigure xserver-xorg
```

驅動程式安裝後，讀者可以用「Ctrl」+「Alt」+「Backspace」來重開你的 X server，若你看到一個很大的 Nvidia Logo，就是安裝成功啦！有時候因為核心模組載入的問題，會需要重新開機。


當讀者開機的時候 Xwindow 開不起來，錯誤訊息告訴你是因為顯示卡驅動的關係，那麼你可以先用「Ctrl」+「Alt」+「F1」進入文字介面，再透過以下方式關閉驅動程式並重新啟動 Xerver。

關閉驅動程式：

```

sudo nvidia-glx-config disable
# 或者讀者也可以把 xorg.conf 的 nvidia 改回來 nv，就不用上面那個動作！

```

 重新啟動 X server:

```
sudo /etc/init.d/gdm restart
```

這時候就會以預設的驅動程式進入 Xwindow，此時你可以開始 debug 了，通常會有錯誤原因是 kernel 版本和 nvidia 模組版本不一樣，讀者可以檢查一下！

6.2.3 AMD (ATI) 顯示卡驅動程式安裝

在截稿時，筆者反覆思考後，決定把這一個部份拿掉了！AMD (ATI) 提供的官方版本 driver 非常不穩定，筆者自己在很多機器上都會出現問題。最近 AMD 併購了 ATI，而 AMD 的政策是比較親近開放原始碼，目前謠傳 AMD 已經將 ATI 原來的顯示卡硬體規格，提供給從事 X Window System 實做開發的 Xorg 團隊來撰寫驅動程式，所以原來充滿 bug 的 fglrx 驅動程式勢必會被拋棄。另外一個原因是 fglrx 驅動程式目前不支援 AIGLX，所以也無法在 Linux 上使用 beryl + AIGLX 跑 3D 桌面。所以目前要在 AMD (ATI) 的卡上跑 3D 桌面，只能透過開放原始碼版本的驅動程式。

目前開放原始碼的 AMD (ATI) 顯示晶片組的驅動程式已經有不錯的效能，至少在筆者的 ibook 上可以透過開放原始碼的驅動程式跑 AIGLX + beryl 的 3D 桌面，且謠傳 AMD 將會支持這個驅動程式，以後的效能表現應該會更好。目前開放原始碼的驅動程式比 fglrx 官方的驅動程式效能還要差，但是在相容性上比較好，bug 也少很多，所以這裡筆者將會改成介紹開放原始碼版本的驅動方式。

由於這幾件事情都是社群剛發生的事情，如 beryl 從 compiz 分裂，ATI 被 AMD 收購等等，所以這裡都是截稿前夕緊急修改的，若有疏漏之處，就要請讀者諸君多多包涵了。

由於 AMD (ATI) 開放原始碼版本的 3D 加速驅動程式安裝方式在第 20 章「Berl 3D Desktop」有介紹到，所以這裡就不再贅述了。讀者可以直接參考該章的安裝方式設定。

6.3 印表機的設定

如圖 6.19，從選單上的系統→管理→正在列印進去設定吧！然後就直接點新增印表機即可。

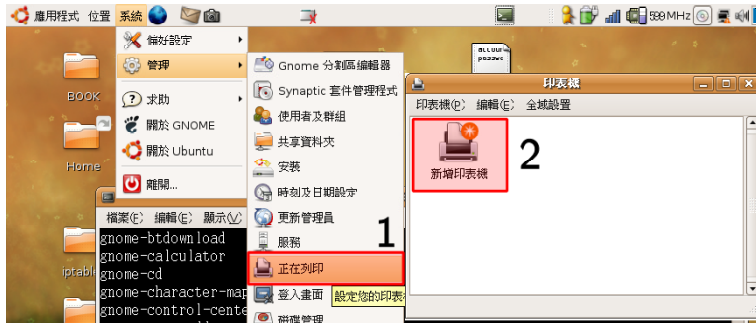


圖 6.19

接下來應該會自動抓到你印表機，如圖 6.20，直接點下一步就可以了！這裡也可以讓你安裝網路印表機，如系統提示操作即可。



圖 6.20

如圖 6.21 接下來選擇你的印表機所需要的驅動程式型號，選擇後，直接下一步吧！



圖 6.21

之後的操作就如安裝系統給你的提示設定，等會還可以列印測試頁喔！安裝完後，你就可以用 Firefox 或者 OpenOffice 等程式列印你要的文件了！是不是很簡單呢？通常連安裝光碟都不需要放進去呢！

6.4 調整終端機 (terminal)

身為 Linux hacker，一定會開機就先開個 terminal 開始敲敲打打，就算沒有視窗介面也是怡然自得。所以把 terminal 弄到良好的環境是對工作效率有很大的幫助。這裡筆者開始介紹如何把文字終端機弄得很舒適。

6.4.1 設定在視窗底下的 GNOME 終端機

這是筆者最常用的 terminal 之一，如圖 6.22，從應用程式→附屬應用程式→終端機就可以執行。筆者習慣拉到快捷列放，這樣要啟動就很快捷唷！

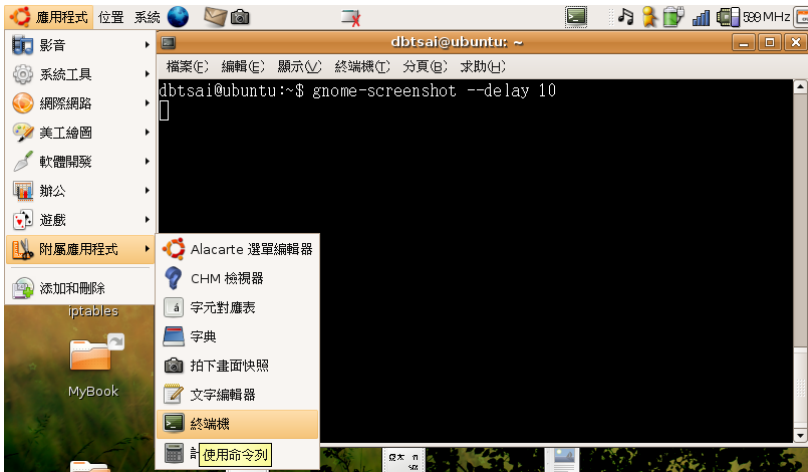


圖 6.22

它預設的字體筆者覺得不好看，筆者是習慣把它調整到大一號字體，並且是黑底白字。讀者可以從編輯→目前的設定組合進去調整。其他還有像背景半透明，或者用底圖的設定，這就給讀者自己品味啦！



圖 6.23

6.4.2 設定 shell 顏色

大部分的 Linux 發行版的 bash shell 登入後都是長這樣的「dbtsai@ubuntu:~\$」，也就是「帳號@hostname:目錄\$」，或許你會嫌太單調了，其實這個是可以調整唷！整個設定的環境變數放在 PS1 裡面。

原始的 Ubuntu PS1 環境設定可以透過以下指令來得知。

得知 PS1 環境變數：

```
dbtsai@ubuntu:~$ echo $PS1
\u@\h:\w\$          # \u 代表 username，\h 代表 hostname，\w 代表目錄
```

所以依照上面的邏輯，你就可以很簡單的去修改看看啦！

簡單的修改 PS1 環境變數：

```
dbtsai@ubuntu:~$ PS1="[\u@\h \W \$] "
[dbtsai@ubuntu ~ $]
```

聰明的你應該發現了，在宣告環境變數時，它只會把整個環境變數的控制指令取代掉，其他的都照常顯示出來。


現在要進階一點，加上顏色控制的功能。我們從一個很簡單的範例開始，例如「\[\033[1;33m\]」這樣的控制碼，就是定義從這之後所有文字全部都是黃色。而 33m 就是定義黃色。我們還可以使用紅色（31m），綠色（32m），藍色（34m），紫色（35m），青色（36m）和白色（37m）。當然還有其他可以用的顏色碼，或背景顏色碼等等，有興趣的讀者可以去查詢 bash 相關資料。

所以我們試試看下面範例的效果吧！

彩色 PS1 環境變數第一版：

```
dbtsai@ubuntu:~$ PS1='\[\033[1;33m\]\u@\h\w$ '
```


讀者會發現，全部都變成黃色了耶！連打的字都是黃色的！真 Orz 阿！這是因為最後我們沒有把它再變回成白色字，所以最後再加上白色的控置碼吧！

 彩色 PS1 環境變數第二版：

```
dbtsai@ubuntu:~$ PS1='\[\033[1;33m\]\u@\h\w$\[\033[1;37m\] '
```


這時候看起來正常多了！只有前面那串 shell 部份是黃色的，打出來的字是正常的白色！各位駭客一定不會那麼容易滿足吧！筆者提供我花俏的設定給讀者嘗試看看，礙於印刷的關係，無法讓讀者直接看到效果，讀者一定要試試看唷！你將會愛上你的 shell 的！

由於排版的關係，無法打成一行，實際實驗的時候要打成一行。

 彩色 PS1 環境變數完成版：

```
PS1='\[\033[1;33m\]\u\[\033[1;37m\]@\[\033[1;32m\]\h\[\033[1;37m\]:\[\033[1;31m\]\w \[\033[1;36m\]\$ \[\033[0m\]'
```

若你要個人使用的話，可以加到 ~/.bashrc 裡面的最後一行，這樣每次開 shell 就會自動定義好色彩。所謂的 ~/ 就是你讀者個人家目錄，例如在 shell 裡面，打 ~/ 會直接用家目錄取代，這樣方便多了。

 編輯個人的 bashrc：

```
gedit ~/.bashrc 或者 gedit /home/帳號名稱/.bashrc
```

若你希望所有人預設都看到，很對不起，筆者還沒找到方法！因為當你把設定寫在 /etc/bash.bashrc 裡面，雖然每個 user 登入都會讀取的，但是 PS1 在 Ubuntu 內每個使用者的 ~/.bashrc 都在再重新定義一次，所以設定還是會被最後讀取的 ~/.bashrc 蓋過去而失效！所以建議自己改想要用的帳號吧！強迫中獎也不是 Linux 的精神哩。

6.4.3 設定 shell 底下常用指令縮寫

重度的 shell 使用者大概會厭煩打多餘的字吧！所以筆者都會設定一些常用的 alias，來增加工作效率。所謂的 alias 就是別名，如範例中，我們設定 alisa ll = `ls -l`，那之後只要打 ll 就可以取代後面 ls -l 那串指令啦！

 編輯個人的 bashrc：

```
gedit ~/.bashrc 或者 gedit /home/帳號名稱/.bashrc
```

✎ 個人 .bashrc 範例

```

if [ "$TERM" != "dumb" ]; then      eval "`dircolors -b`"
    alias ls='ls --color=auto -h'    # 這裡加上 -h 可以自動換算單位到大單位唷！
    alias dir='ls --color=auto -format=vertical'
    alias vdir='ls --color=auto --format=long'
fi

# some more ls aliases
alias ll='ls -l'                    # 使用完整列印模式
alias la='ls -A'                    # 列出所有檔案和目錄，包含隱藏檔
alias l='ls -CF'                    # 分類列出不同屬性的檔案和目錄
alias lm='ls -al | more'            # 使用 more 來分頁

```

如以上範例，把有註解的地方拿掉，改成和上面一樣吧！當然你也可以設其他指令的 alias，就加在後面就好。

6.4.4 設定 framebuffer 解析度和底下的中文終端機

筆者要提醒的是，並非所有的顯示卡都支援使用 framebuffer 輸出，不過市面上常見的顯示卡，如 ATI，Nvidia，和 Intel 整合繪圖顯示卡等都應該有支援，而且 Ubuntu 預設有打開核心的 framebuffer 功能喔！

在伺服器中，為了節省資源，很多人都不喜歡裝上 Xwindow。但預設在純 console 底下的 framebuffer 文字模式，解析度只有 640x480，這麼小的解析度在現在都用 LCD 的時代不夠，所以我們可以透過在 Grub 裡面的參數調整開機後的文字模式解析度。

表 6-5 不同的解析度和色彩數所對應的設定碼

色彩數/解析度	640x480	800x600	1024x768	1280x1024
256	0x301	0x303	0x305	0x307
32K	0x310	0x313	0x316	0x319
64K	0x311	0x314	0x317	0x31A
16M	0x312	0x315	0x318	0x31B

要修改 framebuffer 的解析度，需要修改到 Grub bootloader 的參數，設定檔在 /boot/grub/menu.lst 裡面。

編輯的 Grub 設定檔：

```
gedit /boot/grub/menu.lst
```

設定檔 menu.lst 範例

```

... (略) ...
title          Ubuntu, kernel 2.6.15-26-k7
root           (hd0,1)
kernel         /vmlinuz-2.6.15-26-k7 root=/dev/hda3 ro quiet splash vga=0x31B
initrd         /initrd.img-2.6.15-26-k7
savedefault
boot
... (略) ...

```

如以上 menu.lst 範例，讀者找到了類似的設定部位後，在 kernel 最後面參數加上 vga=0x31B，這樣就會用 1280x1024 16M color 的模式之 console 介面。讀者若要用其他解析度，可以參考表 6-5。修改後，重新開機就可以生效了。

若你常常在「Ctrl」+「Alt」+「F1」純文字終端機工作，應該常常在操作系統指令的時候遇到中文訊息亂碼吧！除了用 export LC_ALL=C 來把所有輸出的訊息改成英文以外，還可以透過 jfbterm 在文字介面下顯示中文，筆者的伺服器都會安裝，尤其是在沒有 Xwindow 的情況下特別方便，它可以顯示 UTF-8 的中文檔名，有點像 DOS 時代的倚天中文。

安裝 jfbterm：

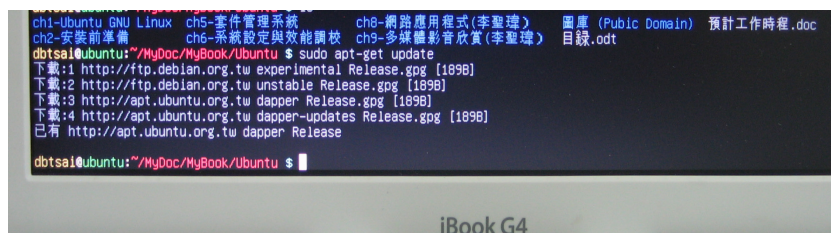
```
apt-get install jfbterm # 這包在 universe，若不能裝，檢查 sources.list 吧！
```

當你進到 tty1 也就是按「Ctrl」+「Alt」+「F1」，用你的帳號登入後，只要輸入以下指令，就可以看到 UTF-8 中文訊息。

執行 jfbterm：

```
jfbterm
```

配合 jfbterm 和 framebuffer 解析度設定效果如圖 6.24。



```

ch1-Ubuntu GNU Linux  ch5-套件管理系統  ch8-網路應用程式(李聖璋)  圖庫 (Public Domain)  預計工作時程.doc
ch2-安裝前準備  ch6-系統設定與效能調校  ch9-多媒體影音欣賞(李聖璋)  目錄.odt
dbtsai@ubuntu:~/MyDoc/MyBook/Ubuntu $ sudo apt-get update
下載:1 http://ftp.debian.org.tw experimental Release.gpg [189B]
下載:2 http://ftp.debian.org.tw unstable Release.gpg [189B]
下載:3 http://apt.ubuntu.org.tw dapper Release.gpg [189B]
下載:4 http://apt.ubuntu.org.tw dapper-updates Release.gpg [189B]
已有 http://apt.ubuntu.org.tw dapper Release
dbtsai@ubuntu:~/MyDoc/MyBook/Ubuntu $

```


圖 6.24

6.5 調整硬碟與光碟效能 hdparm

Ubuntu 基於相容性的考量，IDE 硬碟預設是沒有完全最佳化的。例如 DMA 沒有開，或者用 16bit I/O 等等。所以我們現在教大家來惡搞一翻！但有可能太過惡搞，導致資料損毀或者當機。不過筆者只有遇過當機，沒有資料損壞過，所以在一般的情形下應該還是沒有問題的，但若出現什麼錯誤，風險需要自己承擔哩！這些設定技巧只針對 IDE 硬碟，對於 SCSI 和 SATA 因為預設有開啟 DMA，所以沒有這些選項可以用，不過皆可以透過 hdparm 來測速！

所謂的 DMA 傳輸是當資料要傳到記憶體時，可以直接從硬碟拷貝過去，不需要經過 CPU，所以當然效率好，也可以省下很多 CPU 資源。

我們先看看 IDE 硬碟目前的設定。記得喔！參數是加硬碟的裝置代號，筆者有在第二章討論過了。

 取得硬碟組態：

```

dbtsai@ubuntu:/$ sudo hdparm /dev/hda
/dev/hda:
multcount      = 0 (off)
IO_support     = 0 (default 16-bit)
unmaskirq     = 1 (on)
using_dma      = 0 (off)
keepsettings   = 0 (off)
readonly       = 0 (off)
readahead     = 256 (on)
geometry       = 16383/255/63, sectors = 58605120, start = 0

```

現在硬碟至少都有支援 UDMA4 (ATA66) 或者 UDMA5 (ATA100)，而所謂的 UDMA 就是 Ultra DMA，一定有支援 DMA mode。ATA66 和 ATA100 是用比較細的排線，和光碟機用的不同（光碟機通常是 ATA33），不過他們是可以相容的，讀者可以在開機的時候從 BIOS 得知硬碟是不是 ATA66 以上。

表 6-6 hdparm 常用的設定參數

multcount	-m	這個值決定了在每一次 I/O 動作中，可以掃過的磁區多寡，而幾乎所有的現代化硬碟都已經有支援了。啟用的話，大致上可以增加 5% 到 50% 的匯流排流量。通常都是設為 -m16 以取得最佳效能，而 -m0 為關閉不用。
IO_support	-c	各個選項控制了硬碟晶片組到 PCI 的傳輸模式。幾乎所有的 IDE 裝備都能支援 32 位元同步傳輸了，也有蠻多能支援 32 位元非同步傳輸，非同步傳輸在理論上是比同步傳輸效能好。預設是 16 位元傳輸，參數為 -c0。32 位元非同步傳輸參數是 -c1，同步傳輸參數是 -c3。
using_dma	-d	這是一定要開的啦！ -d0 關掉 DMA 傳輸，-d1 打開 DMA 傳輸。
UDMA 額外參數	-X	ATA33 是 -X66，ATA66 是 -X68 而 ATA100 是 -X69。
cache 讀取測試	-t	測試從硬碟 cache 讀取資料的速度，通常會比 -T 快很多很多，幾乎是實體匯流排的速度。
硬碟讀取測試	-T	測試從硬碟真正讀取資料速度。
硬碟組態詳細資訊	-i	顯示硬碟詳細的組態設定值，如在哪一個 UDMA mode 等。

在還沒有更變硬碟參數以前，我們先來測試個效能看看。

測試硬碟效能：

```
dbtsai@ubuntu:/ $ sudo hdparm -tT /dev/hda
/dev/hda:
Timing cached reads:   848 MB in  2.00 seconds = 424.02 MB/sec
Timing buffered disk reads:  8 MB in  3.79 seconds =  2.11 MB/sec
```

筆者依照表格的資訊，設定硬碟為以下參數，然後再測試一次吧！

設定新的參數並且測試效能：

```
dbtsai@ubuntu:/ $ sudo hdparm -c1 -m16 -d1 -X68 /dev/hda
/dev/hda:
Timing cached reads:   928 MB in  2.00 seconds = 463.10 MB/sec
Timing buffered disk reads:  66 MB in  3.03 seconds =  21.78 MB/sec
```

效能好多了吧！筆者的參數幾乎是每一台機器都可以套用的，所以你可以直接拿去用沒關係。若是你的硬碟是 ATA100 的話，還可以改成 -X69，應該有更好的效能。而光碟機也可以用相同的方法最佳化，不過建議只開啟 DMA 就好，若你膽子很大也可以自己嘗試其他參數。

當然讀者會想要每次開機就自動設定好，那麼有兩個方式，第一個是把這個指令寫到 `/etc/rc.local` 裡，在開機最後就會自動執行到了。而在 Ubuntu 裡面有一個專門放 `hdparm` 的設定檔，筆者建議放在那裡，會比較好維護。

它的設定檔在 `/etc/hdparm.conf` 裡面，它有預設的模組化選項可以設定參數，不過筆者覺得反而不直觀。它最後也有提供直接 `command line` 模式，可以直接把 `hdparm` 的參數寫上去。

設定檔 `/etc/hdparm.conf`：

```
... (略) ...
# 其實這裡是使用比較不正式的作法，正式的方式是修改 /dev/hda { }那一欄
# 裡面的參數。但是筆者還是喜歡直接加上自己要的指令，讀者可以自由選擇。
#/dev/hda {
#     mult_sect_io = 16
#     write_cache = off
#     dma = on
#}

# 先把下面註解拿掉吧！再加上想要使用的 hdparm 指令。
command_line {
    hdparm -c1 -m16 -d1 -X68 /dev/hda
}

# 若有兩個以上硬碟，或者是要修改光碟機參數，只要在弄一個新的 command_line
# 段落即可！記得每個段落只能設定一個 hdparm！
command_line {
```

```
hdparm -c1 -m16 -d1 -X66 /dev/hdb  
}
```

預設 `hdparm.conf` 參數會在開機後執行，所以讀者修改完 `hdparm.conf` 後只要重新開機就可以生效，也就是以後每次開機都會啟用 `hdparm` 加速您的硬碟讀取速度。若讀者要確認剛剛修改的參數可以如預期的啟用或沒有錯誤，以下範例將手動從 `hdparm.conf` 讀取參數並啟動。

手動從 `hdparm.conf` 啟用 `hdparm`：

```
dbtsai@ubuntu:~ $ sudo /etc/init.d/hdparm restart  
* Setting disc parameters... [ ok ]  
# 若顯示以上訊息就代表您的 hdparm.conf 設定正確，若有錯誤的話，  
# 檢查一下您的語法是否正確吧！先用文字模式直接去測您要的參數！喔！  
# 接下來再直接貼到 hdparm.conf 裡面。大家最常犯的錯是有兩個硬碟，  
# 寫在同一個 command_line { } 區段，記得每個硬碟要獨立的區段喔！
```


CHAPTER

7

視窗環境

Linux 上的桌面環境允許多元化、可組合或替換的介面呈現，這些桌面環境我們稱為 Desktop Environment。而 Ubuntu 預設使用的是 Gnome，Kubuntu 是 KDE，Xubuntu 是 XFCE。而各種 Windows Manager 都有他們自己的特色，筆者這裡會都稍微介紹一下，若讀者有興趣可以自己裝起來玩一玩。

7.1 X Server 概要

大家常常會把 X Window System 簡稱為 X，而 X Window 有分為 Client 和 Server。因此在 Linux 底下的視窗系統架構和 MS Windows 的差很多，是一個高度模組化且主從架構的系統。X Server 負責了一切需要和硬體溝通的事情，例如繪製各種畫面型狀與上色，以及硬體加速等等，最後在把實際的結果顯示在各種不同的介面上，如我們的螢幕等。

所有在 X Window 跑的應用軟體都稱為 X Client，例如 Firefox、Openoffice 等等，而 Window Manager 也算是一種 X Client，它定義了一些視窗環境的長相，例如按鈕，佈景主題等等。這些 X Client 軟體會透過通訊通道連線到 X Server 去做最後輸出到設備上，所以根據這些概念，X Server 和 X Client 可以是不同電腦。

7.2 各種不同有趣的 X Window Manager

如同前面所說的每種 X Window Manager (以下簡稱 WM) 都有不一樣的風格，在這裡就介紹幾個常見的 DE 或 WM。

1. KDE

在 Kubuntu 底下預設的 KDE 是藍色風格的，接下來就請讀者看幾張圖，看看 KDE 有什麼特色吧！

這是預設的 KDE 桌面，不過筆者自己有修改過了，比較簡化的版本，如圖所見預設的 KDE 桌面上，不論是圖檔、文件檔、甚至是影片檔都可以真實顯示，不過圖片基本上會照原本的縮圖顯示，但是並不會原圖大小顯示，一般文字檔的話則是顯示前幾行的前幾個字，而影片則是會取出片段中的圖當圖示，方便你閱覽。當你移動你的滑鼠指標到下面的功能面板時，則是以動畫的方式顯示出名稱，蠻漂亮的。



圖 7.1 KDE 桌面環境

那麼要是你將滑鼠指標移動到圖上面的話呢，則會顯示大一一點的圖哦！還有一些詳細的資訊，例如：檔名、檔案格式、檔案大小、修改時間、擁有者、權限等等，而又以不同的檔案（指音樂、影片、文字檔、文件檔等等）會有其他更詳細的資訊，請參考下圖以圖檔做為範例。



圖 7.2 顯示檔案資訊

KDE 應用程式選單

接下來讓我們來看看 KDE 的應用程式選單吧！下圖是 KDE 預設的選單列表，也可以修改你的選單，是要以跳出的方式顯示，或是以動畫的方式出現，當然也可以設定陰影和透明度。

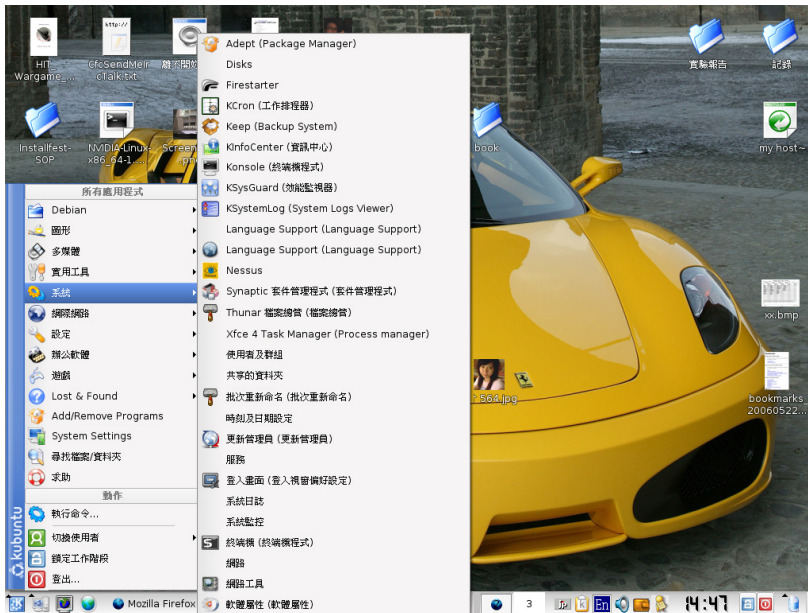


圖 7.3 KDE 應用程式選單

KDE 檔案總管 Konqueror

Konqueror 不僅可以當檔案總管來用，還可以當 Web Browser (網頁瀏覽器)、Samba client (網路芳鄰)，甚至是用來聽音樂也可以，而且還可以將你常逛的資料夾、網頁，加入到 Konqueror 的我的最愛，當然也有 History 功能，可以將你最近逛過的資料夾、網頁或是電腦找出來，也可以當成程式選單的瀏覽器，甚至也可以在同一個視窗開分頁，真的很方便，而且一用就上手。

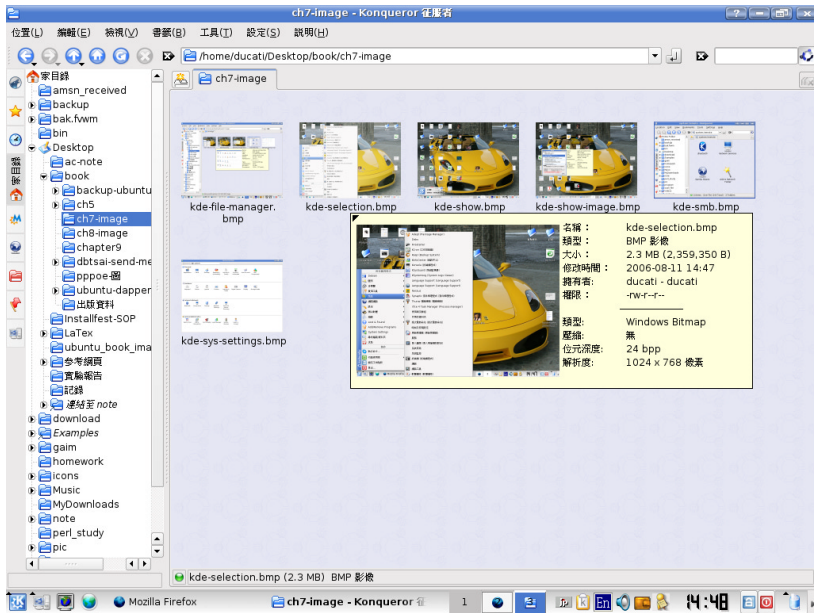


圖 7.4 Konqueror

KDE 控制中心

這個重要的控制中心，可以設定你看不到和看得到的地方。例如：

- **個人設定**；桌面、音效多媒體、選單、個人與隱私、語言、使用者帳號等。
- **硬體**；鍵盤、滑鼠、印表機、電池電源、數位像機、儲存媒體、顯示器等。
- **網路與連線**；藍牙、連接、分享、網路設定、服務等等。
- **系統管理**；日期與時間、硬碟與檔案、字型安裝、登入管理、系統服務、使用者與群組。



圖 7.5 KDE 控制中心

遠端空間

Konqueror 可以與各種裝置連線哦！不管是藍牙、網路服務、Sambashares（網路芳鄰）等，只要跟著步驟設定就可以連上囉！像是 Samba client 也不用什麼設定，只要在圖示點兩下就可以進入網路芳鄰，因為 Samba 是用反組譯的方式，去破解 Windows 的網路芳鄰，所以即使你不知道網路芳鄰的群組 ID，只要同一個網段也一樣可以進到別台電腦所分享的資料夾，除非是有設密碼。

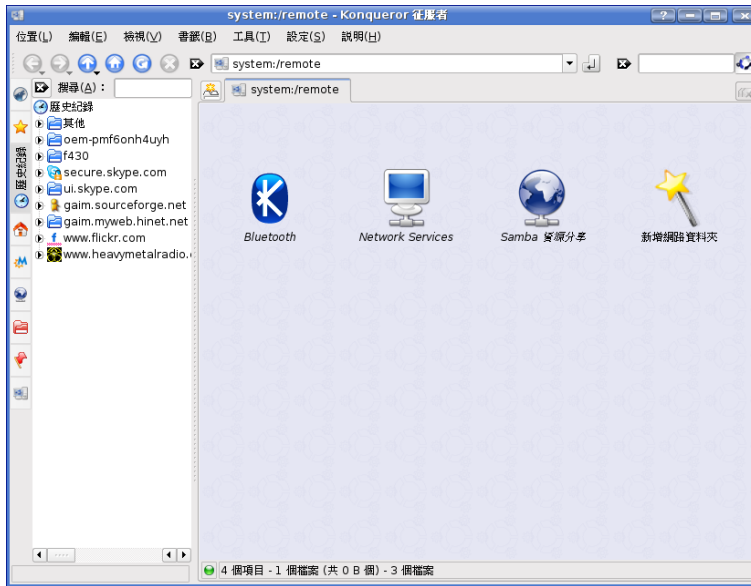


圖 7.6 用 Konqueror 連上各種網路或裝制

儲存媒體

在 KDE 選單旁有個系統選單，不只可以讓你與遠端空間連線，還可以選擇家目錄、使用者目錄，還有儲存媒體可以让你方便管理電腦上的媒體，如硬碟、windows 磁區（如果在安裝時你有掛載上去的話）、甚至是 ipod、隨身碟等。



圖 7.7 管理你的儲存媒體

2. GNOME

Ubuntu 上的 GNOME 的基本配色是橘色，其實 GNOME 看起來有點像是 Mac OS X 的感覺，所以大部份愛用 Apple 的使用者，轉到 GNOME 來也蠻能接受的，不過在這裡就簡單的看一下 GNOME 的桌面配置，在下面的 7.3 章節會有比較詳細的介紹。

GNOME 桌面秀

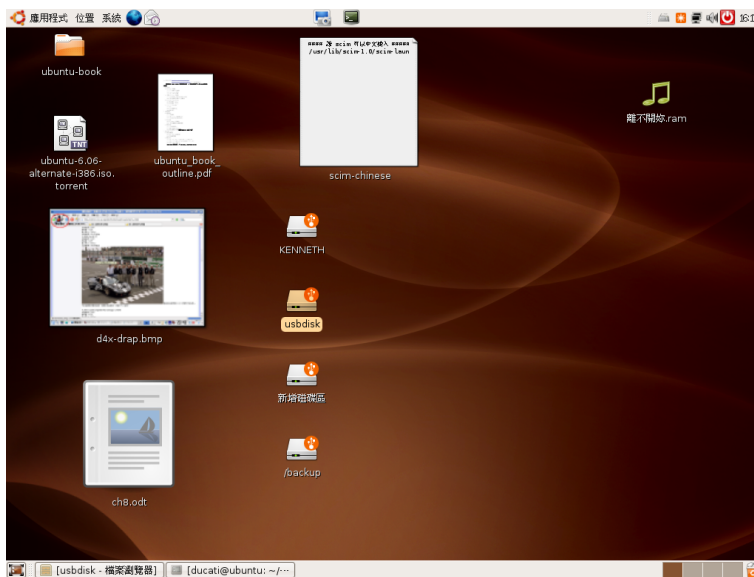


圖 7.8 GNOME 桌面環境

從這張圖來看 GNOME 的配置有點類似 Mac OS X，只不過下面沒有 OS X 的 Dock 而已，如果你想要有類似 Dock 的選單的話也可以自己裝，上面的功能條可以用來放你常用的軟體圖示，以及快速的啟動應用程式、網路位置、Samba client（網路芳鄰），以及管理你的系統或是設定你的桌面環境，來達到適合你使用的環境，以提高生產力。

GNOME 應用程式選單

GNOME 的選單也很方便哦！不論是什麼軟體裝好後都會自動幫你分類。

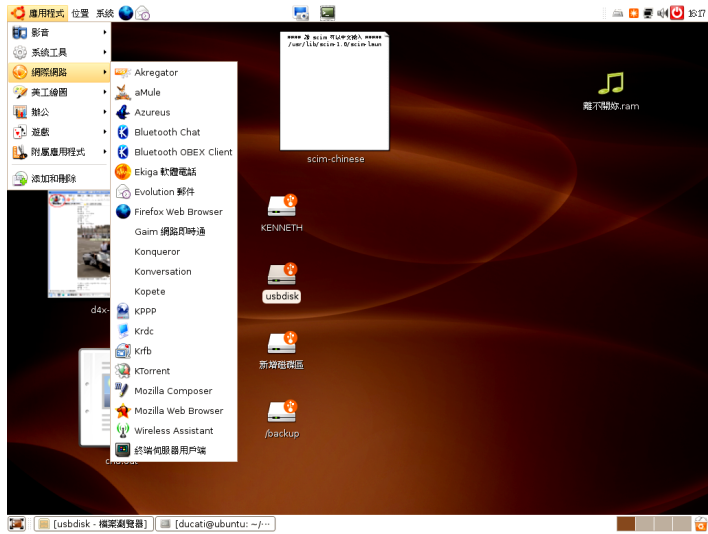


圖 7.9 GNOME 應用程式選單

位置

這個選單可以让你管理個人資料夾、電腦、CD/DVD 製作、Samba client（網路芳鄰），還有所有連接到你電腦上的媒體。

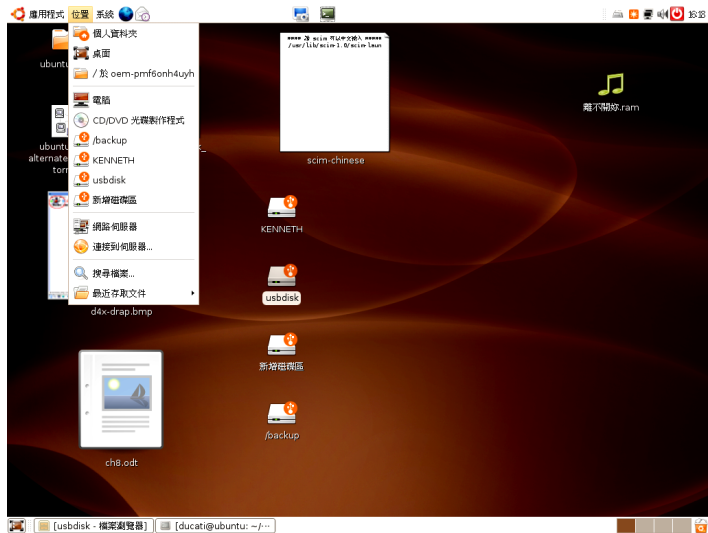


圖 7.10 x 位置

系統

這個選單可以讓你管理設定電腦上的大小事，例如：桌面環境的設定、代理伺服器、電腦硬體的設定，包括電池電源、滑鼠、螢幕解析度、桌面背景、視窗、字型、作業階段、音效、抽取式媒體等等的。管理的話例如：共享資料夾、時間設定、服務、磁碟管理、網路、語言支援、系統監控、以及正在列印等。



圖 7.11 系統管理與設定

檔案瀏覽器

基本上 GNOME 附的檔案瀏覽器可以用來上 Samba client（網路芳鄰），或是預設的 CD/DVD 製作程式也都是用檔案瀏覽器，就可以燒錄光碟了，只要將檔案都放進去，然後有一個燒錄光碟的 icon 按下去就可以了。當然囉！在你瀏覽檔案時不管是圖片或是影片，也都會以預覽圖來呈現哦！

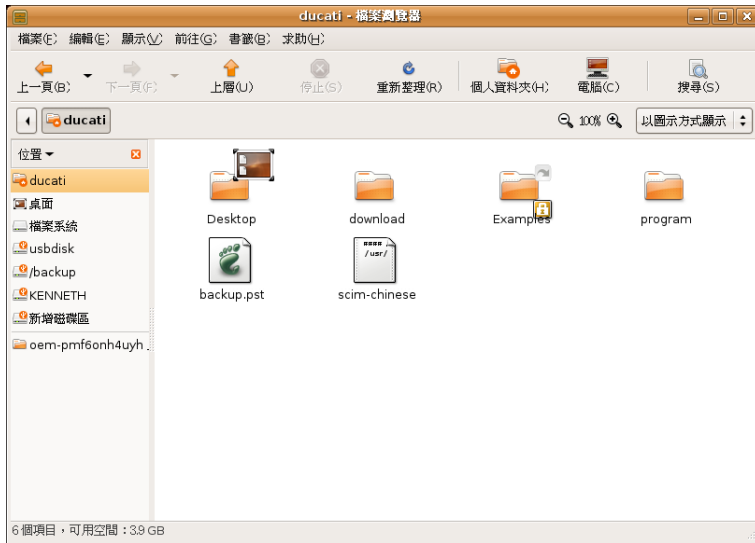


圖 7.12 GNOME 檔案瀏覽器



圖 7.13 GNOME 附的 CD/DVD 燒錄程式

3. XFCE

XFCE 是一個完整又輕便的 Desktop Manager，其實 XFCE 有點像是 GNOME 的簡化版，不過一點也不會輸 GNOME，該有的功能一樣都沒有少，讓我們來簡單的看一下 XFCE 桌面環境吧！

XFCE 桌面秀

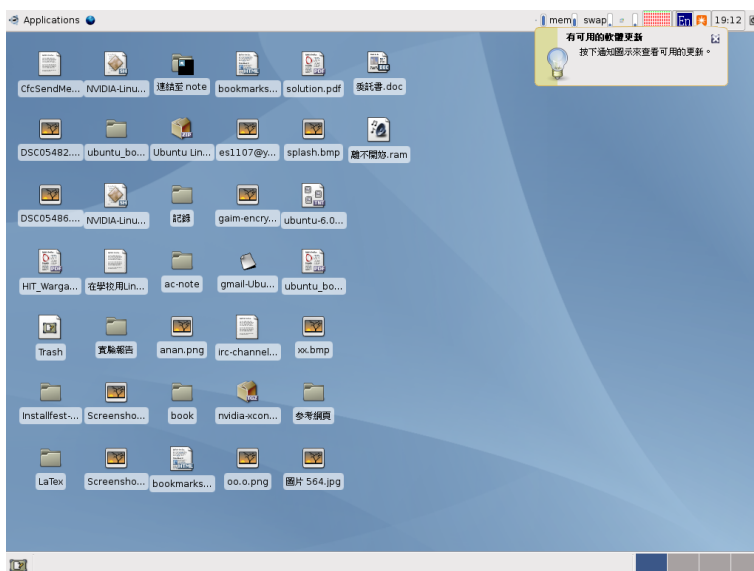


圖 7.14 XFCE 桌面環境

如圖所見 XFCE 跟 GNOME 真的蠻像的，只不過是比較輕量化。相似的原因是因為，xfce 刻意遵照 gnome 的標準來寫，並且在撰寫時處處刻意相容 gnome。而外觀很搭是因為它們都是 gtk+ 程式，都使用 gtk+ theme，這是 gtk+ 本身的特性。左上角是應用程式選單，選單旁的工具條可以加上你常用程式，方便快速啟動，就像 Firefox 一樣的小圖式，只要將應用程式選單裡的圖示拉到上面去放即可。右邊可以像筆書一樣設置系統監控軟體，隨時知道你的電腦狀況，下方工具條的左方是顯示桌面圖示，中間則是你正在使用的應用程式，右邊則是虛擬桌面選單，下方的工具條一樣可以設置圖示，來啟動程式。

XFCE 應用程式選單

應用程式選單一樣簡潔，而且分類也和 GNOME 差不多。

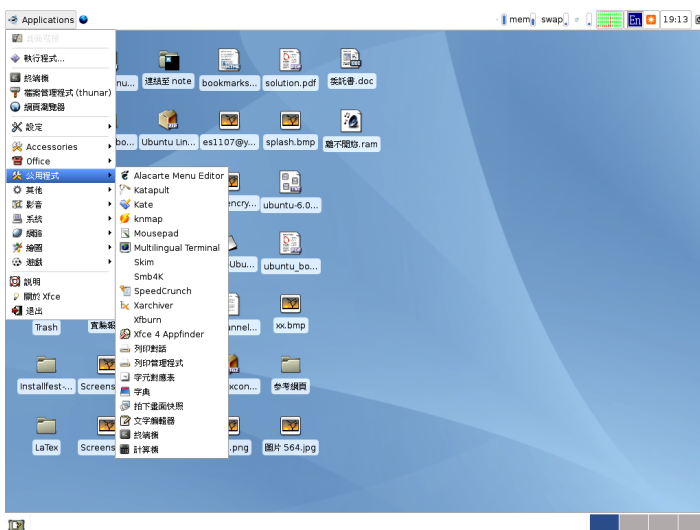


圖 7.15 XFCE 應用程式選單

XFCE 檔案瀏覽器

XFCE 的檔案瀏覽器真的很簡化，功能也比較少，純粹是用來瀏覽檔案的，所以無法上網、燒 CD/DVD，不過啟動速度快，適合用在舊型的電腦上跑。

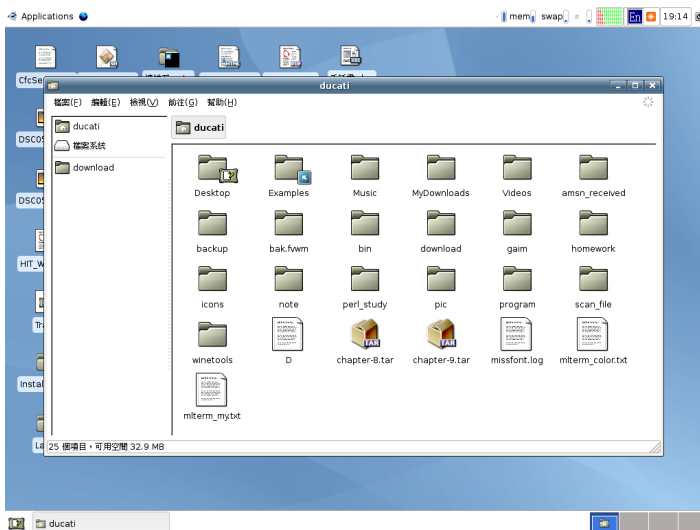


圖 7.16 XFCE 檔案瀏覽器

4. Enlightenment

Enlightenment 是一個很特別的 Window Manager，以超輕量化的介面，應用得恰到好處的虛擬桌面，Enlightenment 的虛擬桌面也是一樣可以設成很多個，甚至是你想要把虛擬桌面攤開來變成橫的或是直的都可以，又或者是一個正方形裡要切成幾個桌面都可以，還有獨特的視窗裝飾，可以將視窗整個收起來變成一條，而且內建就有好幾風格個可以選換，虛擬桌面還可以清楚的顯示你每個桌面的縮圖，可以讓你清楚的看到每個桌面的程式，而不只有框框。另外如果你電腦上同時安裝 GNOME、KDE，Enlightenment 的話，在 GDM 要選擇 session 時、可以發現又多了兩個 session 可以選哦！因為 Enlightenment 可以跟 KDE 和 GNOME 整合，所以又會多出了 E-KDE、E-enlightenment 這兩個 session。

Enlightenment 桌面秀

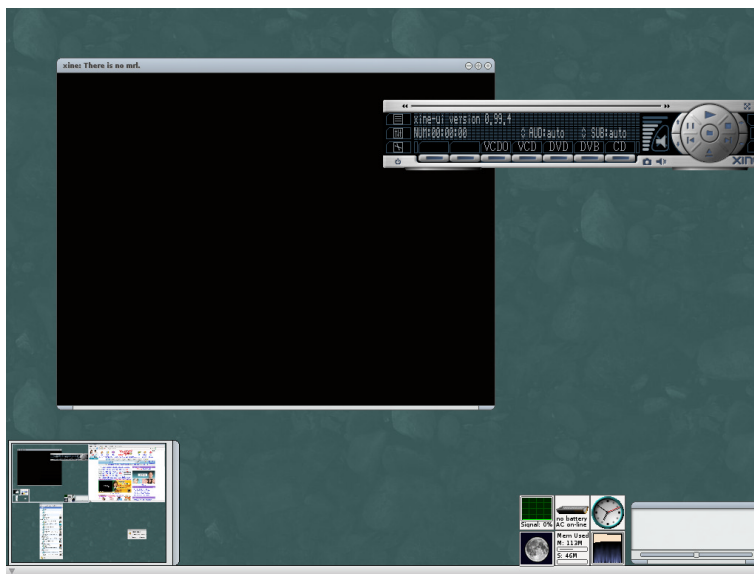


圖 7.17 Enlightenment 桌面環境

Enlightenment 程式

Enlightenment 也有出屬於自己的小程式哦！它有自己出的小時鐘、CPU 監測、網路監測、SWAP 負載監測、系統負載監測，等等的一些桌面小程式。

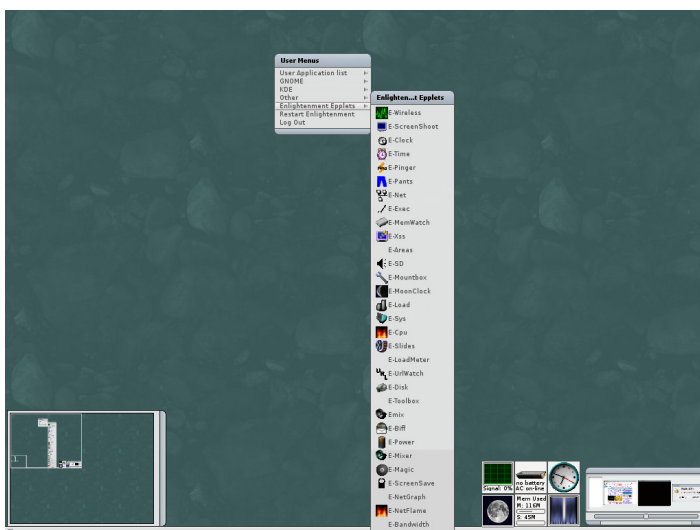


圖 7.18 Enlightenment 小程序

Enlightenment 快速啟動

在 Enlightenment 的桌面上按左鍵，選擇第一項就會出現常用的應用程式選單。



圖 7.19 Enlightenment 常用程式選單

從虛擬桌面來拖曳程式

Enlightenment 可以在左下方的虛擬桌面，按著左鍵拖曳小圖示，可以從任何一個桌面的程式，拖到別的桌面去。

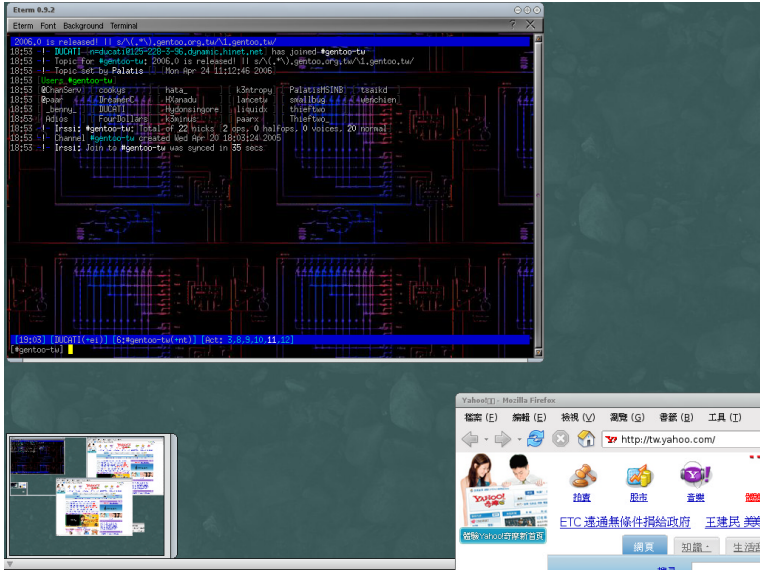


圖 7.20 從虛擬桌面來拖曳程式

5. ICEWM

ICEWM 是一個超簡化的 Window Manager，比 Enlightenment 還要簡化輕量哦！不過預設的桌面環境不是很美，而且要改設定的話一定要直接修改設定檔案。不過還是有人把 ICEWM 改得很漂亮，但是使用 ICEWM 的通常是進階使用者，而且其平日操作大多在終端機環境下完成，加上想要調校成適合自己的環境蠻花時間的，所以平常也鮮少看到有人使用。

ICEWM 桌面秀



圖 7.21 ICE 桌面環境

如圖所見 ICEWM 表面上看上去沒什麼，不過裡面藏的東西可多的咧。

ICEWM 程式選單

ICEWM 可以有自己的選單，也會將 GNOME、KDE 的選單包進去。

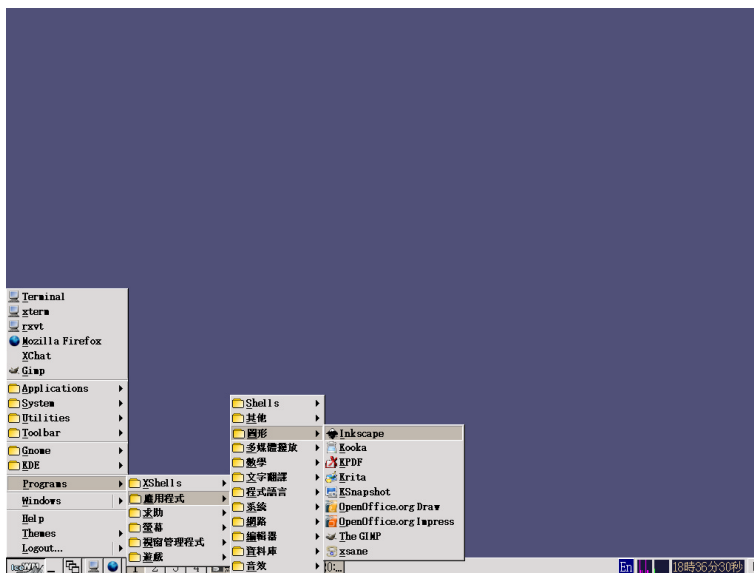


圖 7.22 ICEWM 程式選單

因為篇幅的關係，所以暫時就先不介紹怎麼改造 ICEWM 的桌面環境了，不過若讀者你有興趣的話，也可以自己找找資料，其時 ICEWM 的設定不難，相信在你找資料的同時，會發現很多 ICEWM 的驚人之處。

7.2.1 如何在 GDM 切換不同 X Window Manager

ubuntu 開機後還沒輸入帳號時，在 GDM 按下 F10 後會出現選單，這時候選擇作業階段選項。



圖 7.23 選擇作業階段

然後會出現作業階段選單，這時候選擇你想要的 X Window Manager 即可。



圖 7.24 選擇 X Window Manager

7.2.2 安裝 KDE

由於有 metapackage（虛擬套件）這方便的東西，來幫我們整合 KDE 桌面環境所需的套件，所以我們不必下很多指令來安裝 KDE，只要將 KDE 的 metapackage 下指令裝了以後，只要設定幾個選項就可以將 KDE 裝好了。

安裝 KDE 的 metapackage

```
$ sudo apt-get install kubuntu-desktop
```

大約會下載 133 MB 的檔案，解壓縮安裝後會消耗磁碟約 403 MB 的空間。

安裝過程中遇到的選項

在安裝過後提示你，將開始 KDM 的設定，這時候按 Enter 即可。



圖 7.25 開始 kdm 設定

接下來 KDM 會問你要選擇 KDM 或 GDM，還是選 GDM。



圖 7.26 選擇 GDM

不過依安裝順序不一樣，可能還會遇上其他的問題選項，如：會問你解析度，這就看你的電腦螢幕，可以支援到多高的解析度了，就請自行判斷哦！

7.2.3 安裝 XFCE

和 KDE 一樣只要安裝 XFCE 的 metapackage 就可以囉！

安裝 XFCE 的 metapackage

```
$ sudo apt-get install xubuntu-desktop
```



圖 7.27 XFCE 登入畫面

7.2.4 安裝 Enlightenment

安裝 Enlightenment 也很快哦！只要直接安裝 Enlightenment 即可。

```
$ sudo apt-get install enlightenment
```

安裝時會自動幫你把所有相依的套件都安裝好。



圖 7.28 在 GDM 選擇 Enlightenment 登入

7.3 認識 GNOME 基本操作

基本操作熟的話，可以讓你快速上手哦，如果是新手的話可以看一下。

中鍵即可貼上

在 Linux 的 X 下，只要將你想要的文字用滑鼠選起來，反白後在想貼上的地方按下中鍵，或是滑鼠滾輪即可貼上，不過這種貼上的功能跟 Ctrl+C 不一樣，Ctrl+C 或是按右鍵複製是貼到剪貼簿，而這個功能是快速暫時的，若反白選起來了以後，不小心再選到別的文字，就會失效，只有最後被反白的文字會被貼上。

虛擬桌面

好好善用你的虛擬桌面可是可以讓工作效率大大提升。虛擬桌面的用處是，用電腦的時候有時候會開很多親窗，常常視窗會多到要點來點去才能找到想要的，那麼如果可以依自己的喜好分類呢？例如：上網類放在同一個桌面，聊天類放同一個桌面，工作的事情也另外放一個桌面，這樣是不是可以提高桌面的使用效率，然後另外一個桌面還可以用來電影也不錯吧！

但是要記得每個 X Window Manager 的虛擬桌面特性是不一樣的，讀者有興趣的話，可以每個 WM 都試看看哦！

新增小圖示快速啟動

假如今天我們裝了一個程式，或是想要修改程式的預設值，其實可以直接加在上面或是下面的工具列哦！請看以下示範。首先在上方的工具列按一下右鍵，選加入面板



圖 7.29 加入新啟動小圖示的面板

然後會出現加入面板的選單，請點選自訂程式啟動器

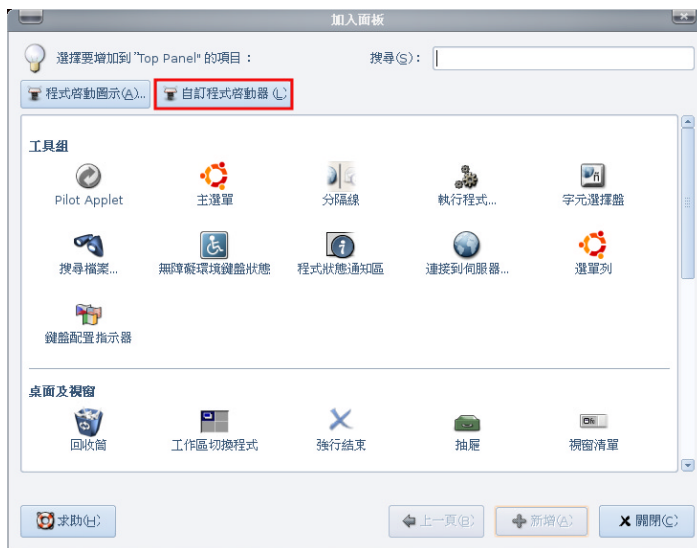


圖 7.30 加入面板

接著會出現建立啟動圖示的選單，這裡筆者以 FileZilla3 做為範例，所以在名稱我們輸入 FileZilla3，其他的一般名稱和備註可以不用輸入，看個人喜好囉！最重要的是指令的地方，指令最好是填上 FileZilla3 執行檔的絕對路徑，例如筆者自己的 FileZilla3 執行檔路徑，/home/ducati/program/FileZilla3/bin/filezilla 填上去，這樣就可以了。



圖 7.31 加入名稱與路徑

不過讀者你應該發現還沒有圖示吧！這時候點選沒有圖示的按鈕，輸入 FileZilla3 的圖示路徑，/home/ducati/program/FileZilla3/share/filezilla/resources/，然後選 FileZilla3 的圖示按確定就可以了。

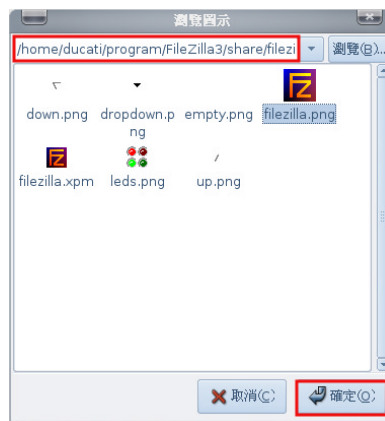


圖 7.32 加入圖示路徑

接著圖示就會跟著出現囉！



圖 7.33 出現圖示

然後看看在面版上是不是多出了新的啟動圖示



圖 7.34 新啟動圖示出現

安全的退出 USB Disk

安全的退出 USB Disk 是很重要的，因為如果沒有安全退出的話，資料很可能不會被存進去，所以用 USB Disk 要退出的時候，請記得要安全退出，筆者自己也常犯這個錯誤，常常沒存到資料就拔出來了。

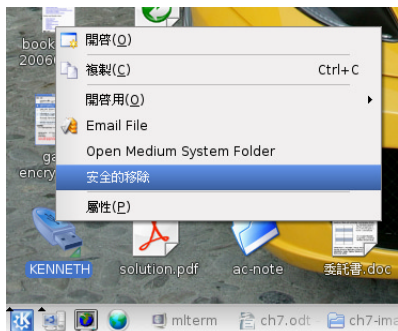


圖 7.35 安全退出 USB Disk

開啟檔案瀏覽器

第一次使用 GNOME 或 KDE 的讀者，一開始可能會找不到檔案瀏覽器，筆者自己第一次用就是這樣，所以在這裡還是說一下，怎麼開啟 GNOME 和 KDE 的檔案瀏覽器。

1. 開啟 GNOME 檔案瀏覽器，位置--> 個人資料夾 或 桌面



圖 7.36 GNOME 檔案瀏覽器

2. 開啟 KDE 檔案瀏覽器，面版上的 系統選單--> 家目錄(要先點開系統選單)



圖 7.37 KDE 檔案瀏覽器

7.3.1 修改登入畫面 (GDM)

其實系統中有幾個可以選擇的登入畫面，不過要是不喜歡怎麼辦呢？當然就上網站去找找囉。在這裡我們介紹幾個網站，是 Linux user 愛用的，可以將你的 X Window Manager 改裝的美美的，而且不管是佈景主題、Splash Screen、GDM 登入畫面、系統圖示、桌面背景等，都可以修改，當然其實還不只有這些，像是滑鼠游標、佈景、字型也可以自由更換。

- KDE-LOOK (<http://www.kde-look.org>)：假如你是 KDE 的愛用者，請看這個網站。
- GNOME-LOOK (<http://www.gnome-look.org>)：假如你是 GNOME 的愛用者，請看這個網站。
- XFCE-LOOK (<http://www.xfce-look.org>)：假如你是 XFCE 的愛用者，請看這個網站。

那麼接下來進入正題，怎麼樣修改 GDM 的登入畫面，你可以選擇系統預付的也可以，或是要上 GNOME-LOOK，尋找自己喜歡的，下面的例子我們用上網找到的 GDM Themes 來試試。

到 GNOME-LOOK 選擇左邊的 GDM Themes

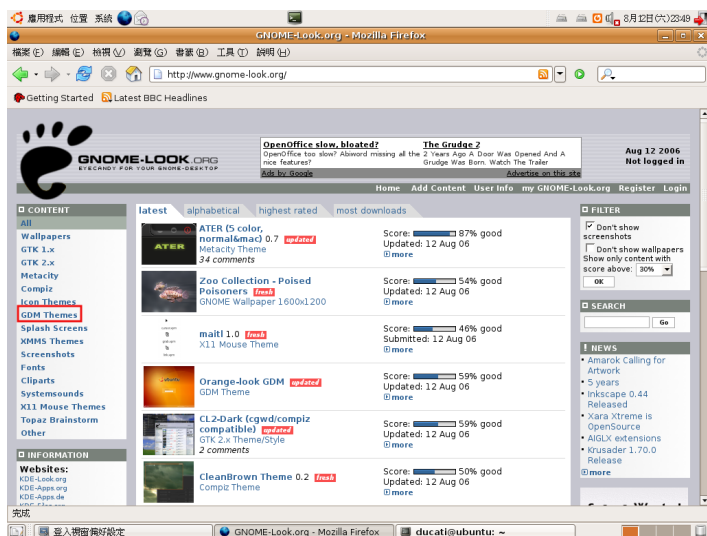


圖 7.38 GNOME-LOOK

1. 上 GNOME-LOOK 找自己喜歡的 GDM Themes

用 Sleek Dragon 這個 GDM Themes 做例子，網址：

<http://www.gnome-look.org/content/show.php?content=25716>

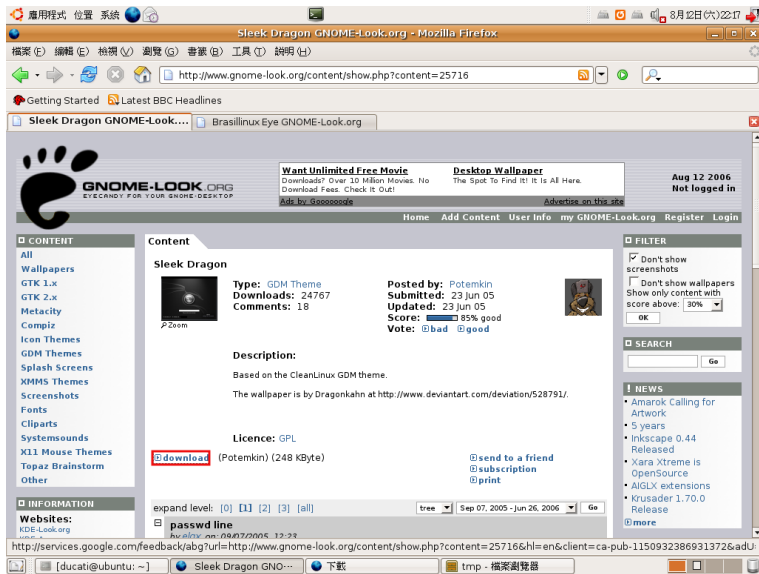


圖 7.39 Sleek Dragon GDM Themes

2. 開始安裝

選擇 download 下載回來，不用解壓縮，先開啟修改登入畫面的工具。系統--> 管理--> 登入畫面。



圖 7.40 登入畫面修改程式

然後可能會要你輸入帳號密碼，如果前沒多久有輸入過一次，就不會再提示你輸入。

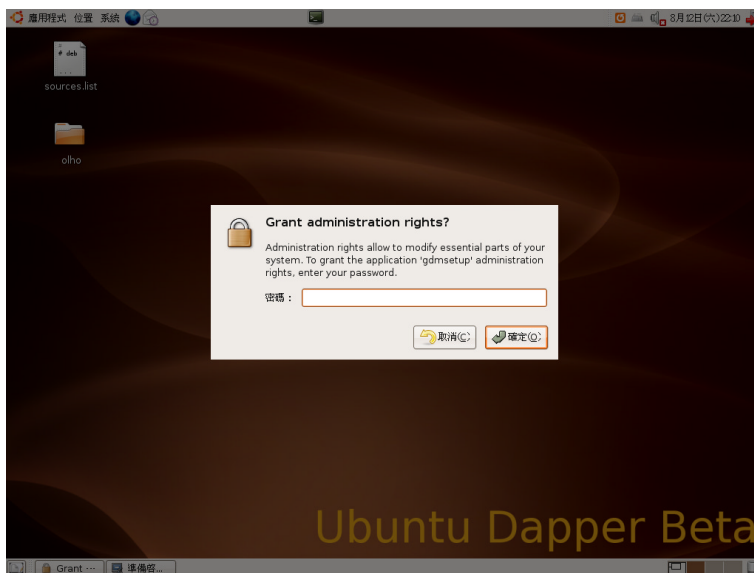


圖 7.41 輸入帳號密碼

輸入後接著登入畫面修改程式就會開啟，這時候我們可以看到，其實預設還有好幾個 GDM Themes 可以選哦！那要怎麼安裝從網路上下載的呢？

首先按一下 Add。



圖 7.42 新增下載的 GDM Themes

選擇剛才下載的 tar.gz 檔案位置



圖 7.43 選擇 GDM Themes .tar.gz 的檔案

選好後可以發現多了新的 GDM Themes，選擇了以後關掉視窗。



圖 7.44 新的 GDM Themes

3. 測試是否安裝成功

接著可以登出看看 GDM Themes 是否被換掉了



圖 7.45 Sleek Dragon GDM Themes

這樣就大功告成囉！其他的 GDM Theme 也可以用同樣的方法。

7.3.2 修改登入過程畫面 (Splash Screen)

首先呢一樣到 GNOME-LOOK 點選 Splash Screen，找到自己喜歡的 Splash Screen 後，可以在檔案旁按右鍵，選擇在此解壓縮。



圖 7.46 解開壓縮檔

然後將檔案 copy 到 Splash Screen 的資料夾

```
$ sudo cp 檔案所在資料夾/xxxx.png /usr/share/pixmaps/splash/xxxx.png
```

接著安裝 gconf，這是為了要修改 Splash image 的設定值

```
$ sudo apt-get install gconf
```

然後在 console 下開啟 gconf-editor

```
$ gconf-editor &
```

然後選擇 apps--> gnome-session--> options，接著在 splash_image 按右鍵，選擇修改設定鍵，然後輸入你剛才 copy 過去 /usr/share/pixmaps/splash/ 的圖檔名稱，像是 splash/xxxx.png 即可。

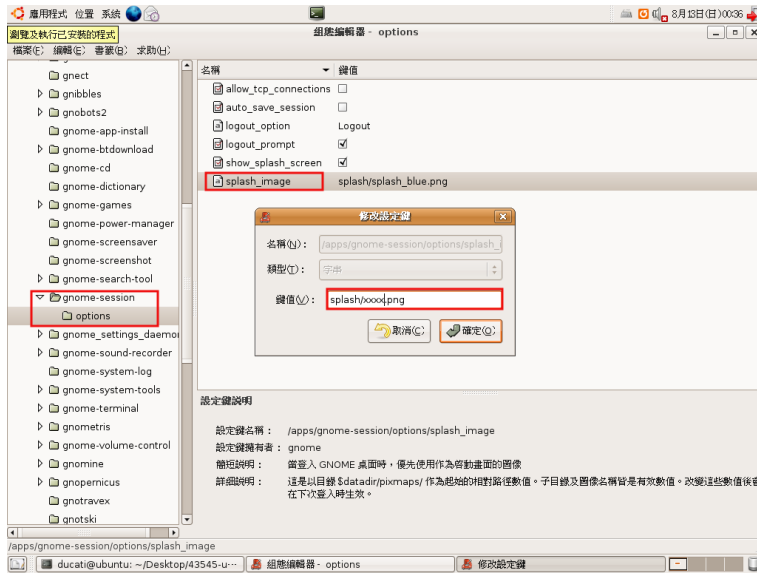


圖 7.47 用 gconf-editor 設定 Splash image

那就登出再登入，看看登入中的畫面是不是有改變呢？

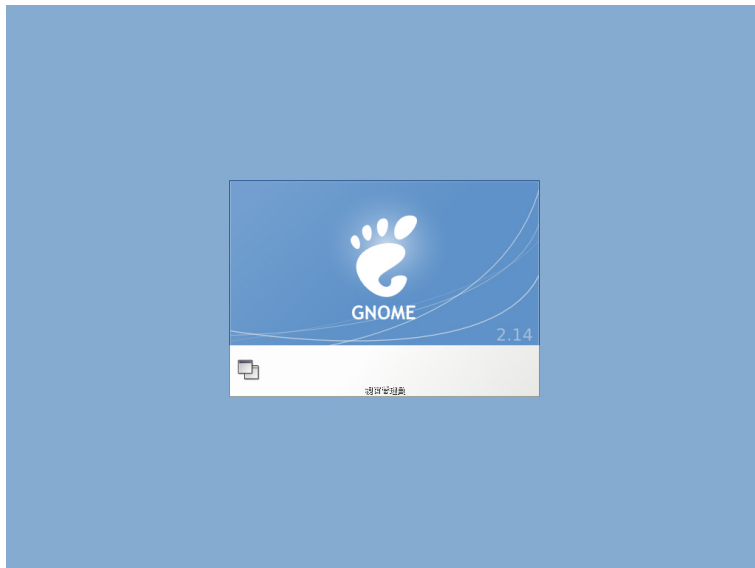


圖 7.48 新的登入中畫面

如果看的新的登入中畫面中畫面就 ok 囉！如果沒有成功的話，有可能是
在 gconf-editor 的地方打錯名稱了，或者是檔案沒 copy 到
/usr/share/pixmaps/splash/ 裡面。

7.3.3 修改佈景主題

1. 修改界面配色

系統--> 偏好設定--> 佈景主題 開啟修改工具，第一個要說的是修改整個介面的
配色。

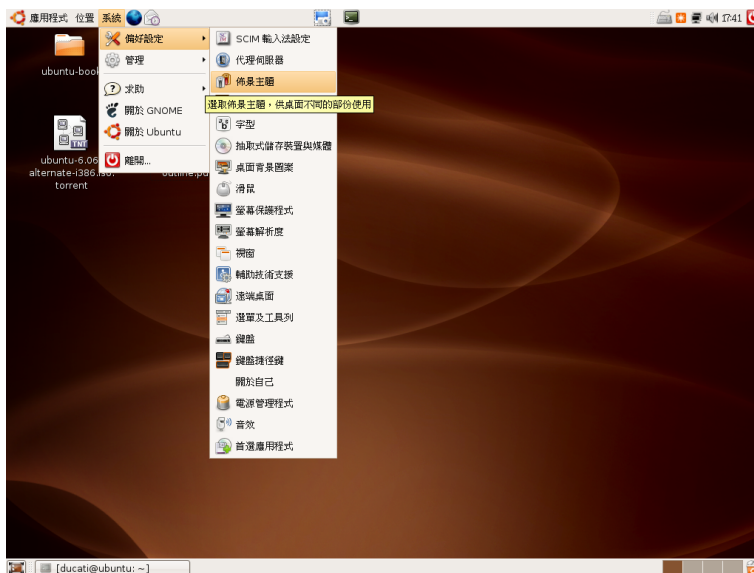


圖 7.49 開啟修改佈景主題

預設就有一些配色可以選囉！也可以自己修改設定值



圖 7.50 修改佈景主題工具

如果不喜歡可以改設定值或是上網下載，網址

<http://art.gnome.org/>

在網址右邊的選單列點選 Application

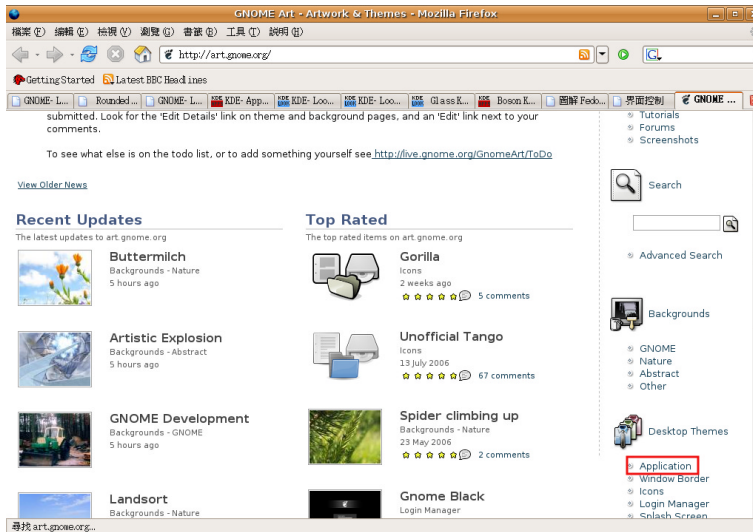


圖 7.51 點選 Application

然後可以看到很多佈景主題配色哦！把喜歡的下載回來

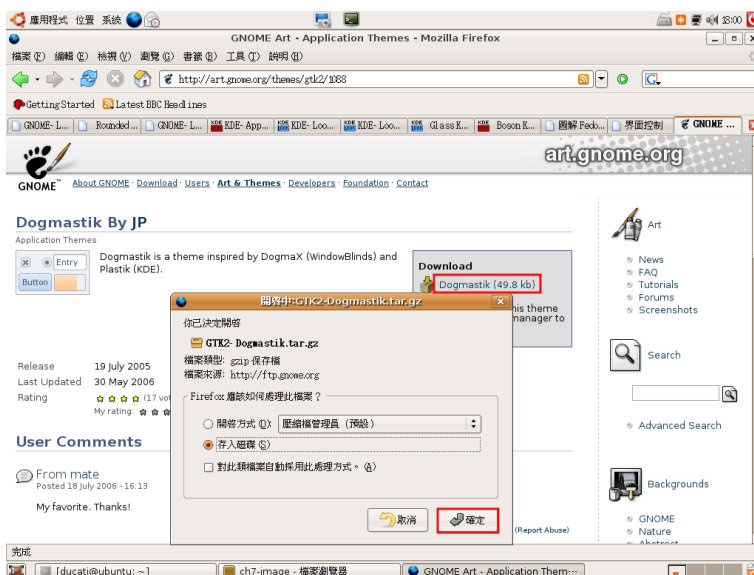


圖 7.52 下載佈景主題配色

接著選擇剛才下載回來的檔案安裝

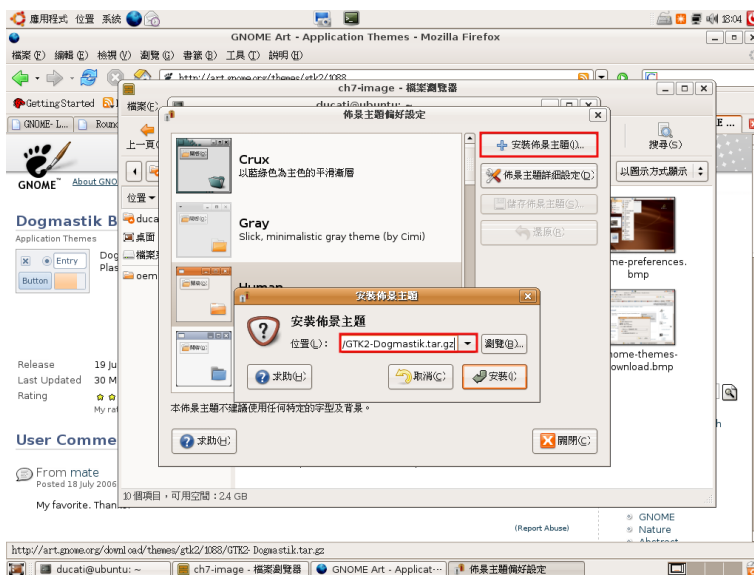


圖 7.53 案裝配色檔案

然後點選佈景主題詳細設定，然後選擇界面控制，找到你剛才所安裝的配色主題吧！

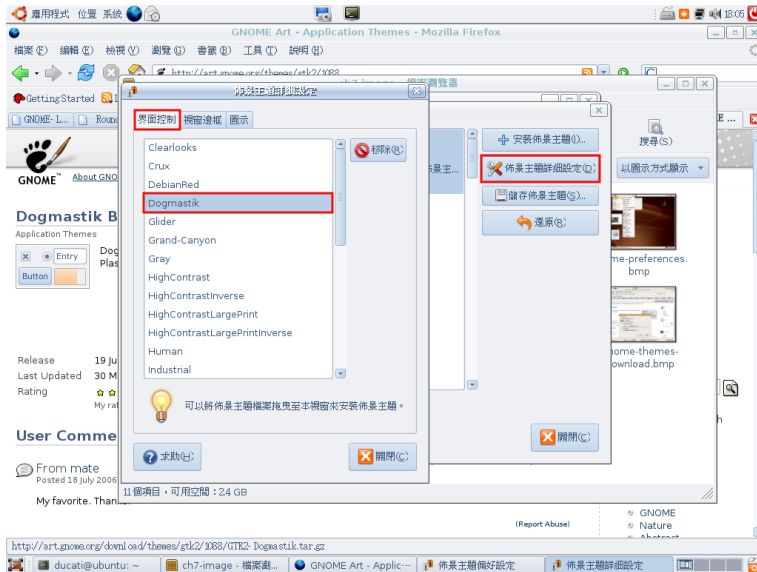


圖 7.54 選擇你的界面控制

這樣就可以囉！

2. 修改視窗裝飾

到 <http://arg.gnome.org/> 選 Window Border

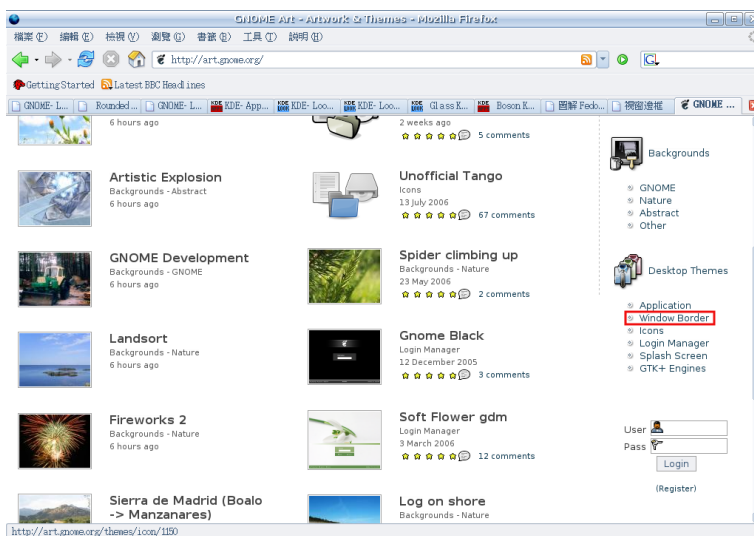


圖 7.55 點選 Window Border

看到喜歡的 Window Border 就下載圖來吧！

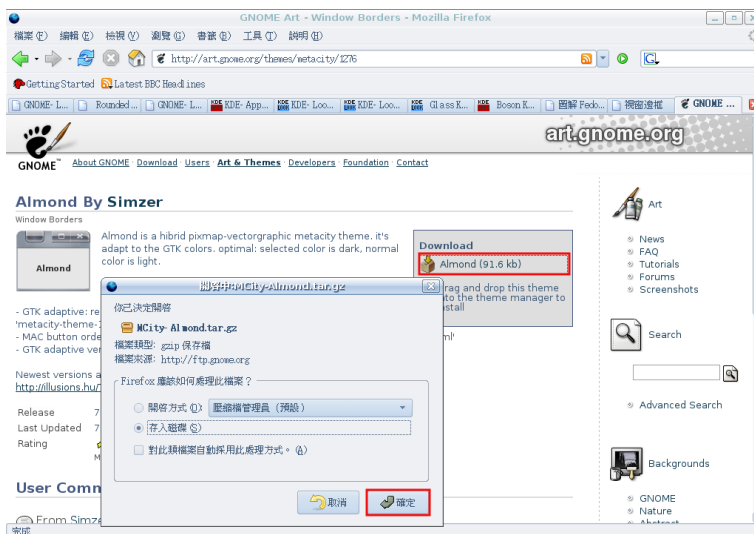


圖 7.56 下載 Window Border

然後一樣開啟佈景主題修改工具 系統--> 偏好設定--> 佈景主題，點選安裝佈景主題，選擇你剛才所下載回來的檔案。



圖 7.57 安裝 Window Border

安裝好後再點選**佈景主題詳細設定**--> **視窗邊框**，選擇你剛才安裝的 Window Border。

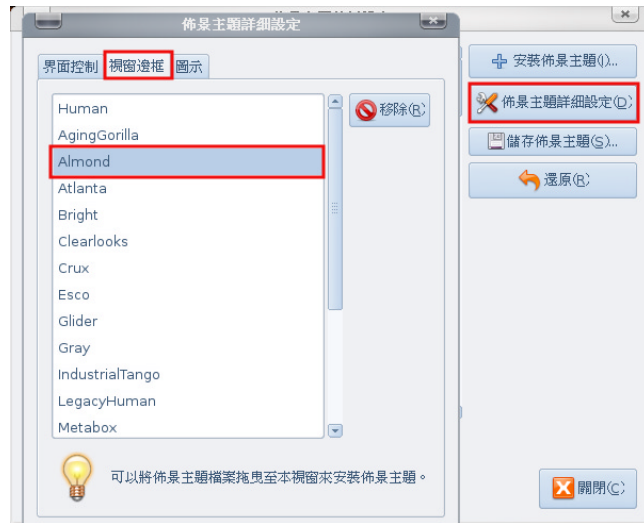


圖 7.58 選擇 Window Border

點下去的同時就可以看到，視窗裝飾已經改變囉！

7.3.4 修改系統圖示

你是否對系統圖示看膩了呢？或者也不喜歡系統上所附的呢？那麼一樣到 <http://art.gnome.org> 下載吧！點選 Icons。

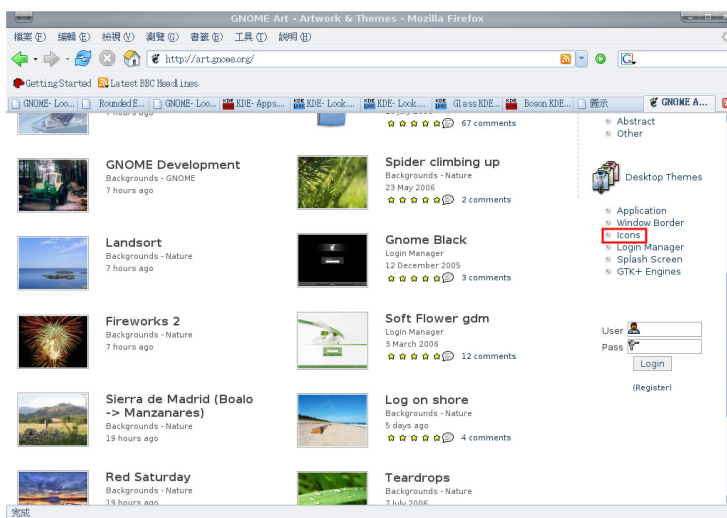


圖 7.59 點選 Icons

接著也下載喜歡的 Icons，點選 download 下面的字樣。

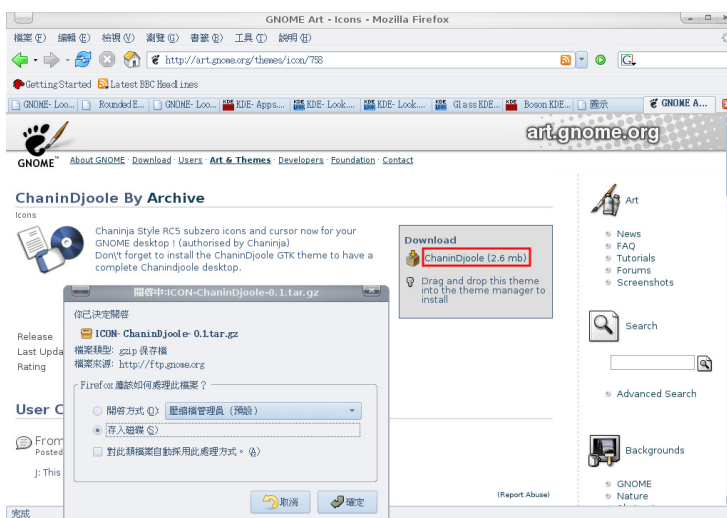


圖 7.60 下載 Icons

下載圖來後一樣開啟系統--> 偏好設定--> 佈景主題，點選安裝佈景主題選擇你剛才所下載回來的檔案。



圖 7.61 安裝新 Icons

接著點選 佈景主題詳細設定--> 圖示，然後選擇你剛才安裝的 Icons 就可以囉！

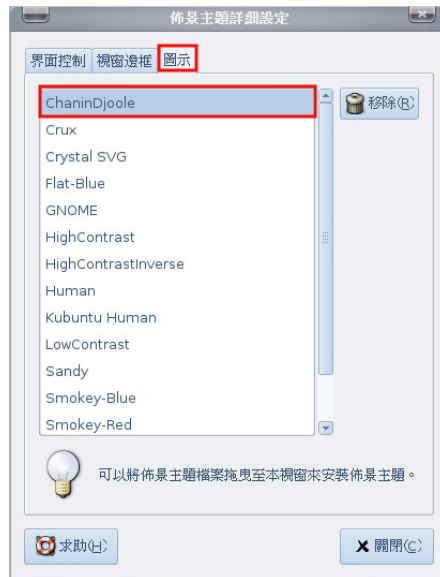


圖 7.62 選擇新圖示

然後開啟選單或是檔案瀏覽器，看看圖示是不是變了呢？

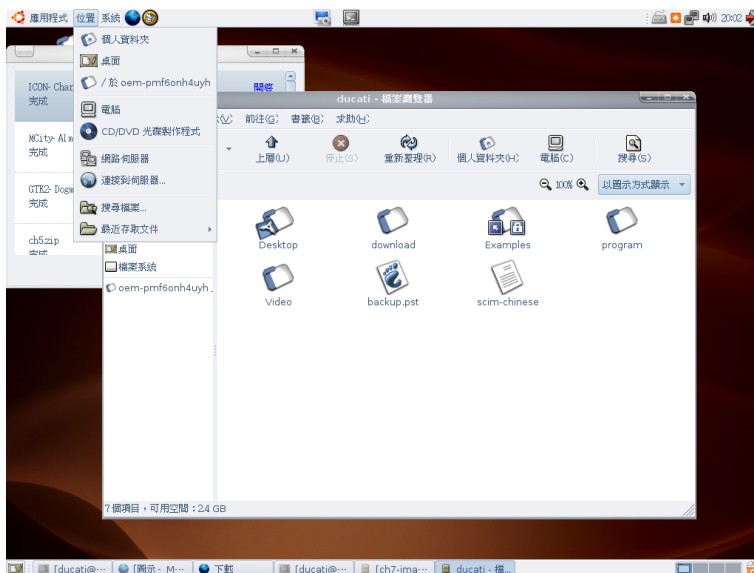


圖 7.63 檢視新圖示

新圖示蠻好看的吧！

7.3.5 修改桌面背景

先在桌面上按右鍵，選更改桌面背景。

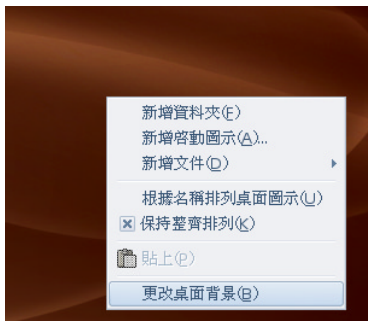


圖 7.64 更改桌面背景

然後在桌面偏好設定選擇加入桌布，將你喜歡的桌布加入即可



圖 7.65 加入桌布

接下來看看桌布是不是新的囉！



圖 7.66 新桌布

7.3.6 安裝桌面小程式 (gDesklets)

這是個很好用的小程式，可以讓你選擇安裝一些小程式放在桌面上，如監測系統的 CPU、RAM、SWAP 的溫度，或是使用的百分比等等的，也有天氣狀況、音樂播放、小型工具面板，這些小工具 gDesklets 都有幫你做分類，只要你安裝了 gDesklets，依照分類就可以找到符合你需求的小工具，只要在小工具的圖示點兩下滑鼠就可以裝上去囉！

1. 安裝並啟動 gDesklets

```
$ sudo apt-get install gdesklets
```

應用程式--> 附屬應用程式--> gDesklets

2. 選擇並安裝小程式

第一次開啟 gDesklets 後，會有一個訊息提示，看完後就可以直接關掉。



圖 7.67 提示訊息

然後就可以看到主程式囉！接著看到左邊是程式的分類表，右邊則是同類的小程式，你可以在喜歡的小程式點左鍵兩次執行，執行後就找個地方擺，沒有擺好的話，在小程式上點左鍵就可以移動或設定。

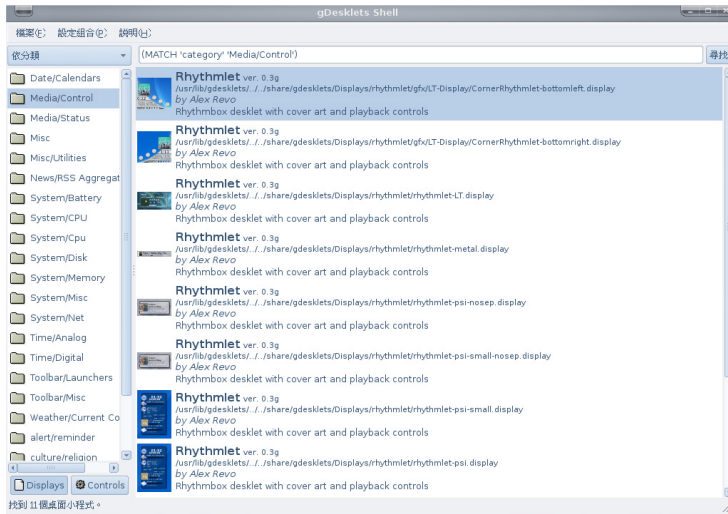


圖 7.68 選擇你喜歡的小程式

那麼接下來看看桌面，是不是變得更帥氣了，怎麼改都喜歡，而且還可以上網找小程式安裝哦！



圖 7.69 改裝好的桌面

還算蠻漂亮的桌面吧！

7.4 中文輸入法安裝設定

在 Linux 下中文的輸入法引擎的安裝，並不是一開始就 ok 的，即使是 ubuntu 一開始也只有裝好 SCIM，但是要如果想試看看別的輸入法引擎呢？那麼如果安裝了很多不同的輸入法引擎，要怎麼設定我想用哪一個，Linux 上又有什麼輸入法？基本上 Linux 常用的輸入法引擎有 SCIM、GCIN、XCIN、OXIM 這幾種，怎麼設定、安裝和使用，就請接下去看囉！

7.4.1 什麼是 im-switch

im-switch 是官方所提供的輸入法引擎選擇系統，可以列出你目前所安裝所有輸入法引擎，也可以自動幫你切換輸入法的語系，使用的方法也很簡單！不過別擔心，筆者在 ubuntu 上測試過了。使用者只需要裝完了 im-switch 之後，在 ubuntu 上會自動幫你載入該輸入法的設定檔，只需要你用 im-switch 來檢視有什麼輸入法，以及切換就可以囉！這個指令不用 superuser 就可以使用了。

安裝 im-switch

```
$ sudo apt-get install im-switch
```

列出已安裝的輸入法

```
$ im-switch -l
```

選擇已安裝的輸入法

```
$ im-switch -s oxim
```

或是

```
$ im-switch -s gcin
```

這指令會幫你在 `~/xinput.d` 建立一個軟連結 (symbolic link)，所以當你進入 X 的時候 im-switch 會幫啟動你所選擇的輸入法引擎，這時候如果你是用 superuser 來輸 command 的話，會變成是整個系統的設定，而不是單一 user 了。

你也可以加上 -z option 來變換語系

```
$ sudo im-switch -z zh_TW -s oxim
$ sudo im-switch -z zh_HK -s oxim
$ sudo im-switch -z zh_CN -s oxim
$ sudo im-switch -z ja_JP -s oxim
```

7.4.2 scim 萬國語言輸入程式

scim 是 ubuntu 預設的中文輸入法，也就是說如果你安裝時可以先 DHCP 取得 ip 的話，那麼 ubuntu 就會自動幫你裝好中文輸入了，連好用的新酷音也是自動裝上去囉！來看看 scim 長什麼樣囉！



圖 7.70 scim

設定 scim 在小圖示上按右鍵，選 **設定 scim**



圖 7.71 選擇設定 scim

scim 的設定視窗。

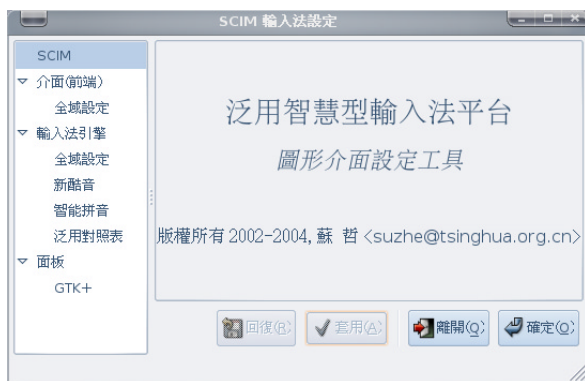


圖 7.72 scim 設定視

接著可以選**輸入法引擎**--> **全域設定**，來設定想要加入或移除的輸入法，只要將想用的輸入法打勾就可以用囉！別忘了設定完要在小圖示按右鍵，**重新讀取設定值**哦，如果新的設定值不管用的話，登出再登入一次就好了。

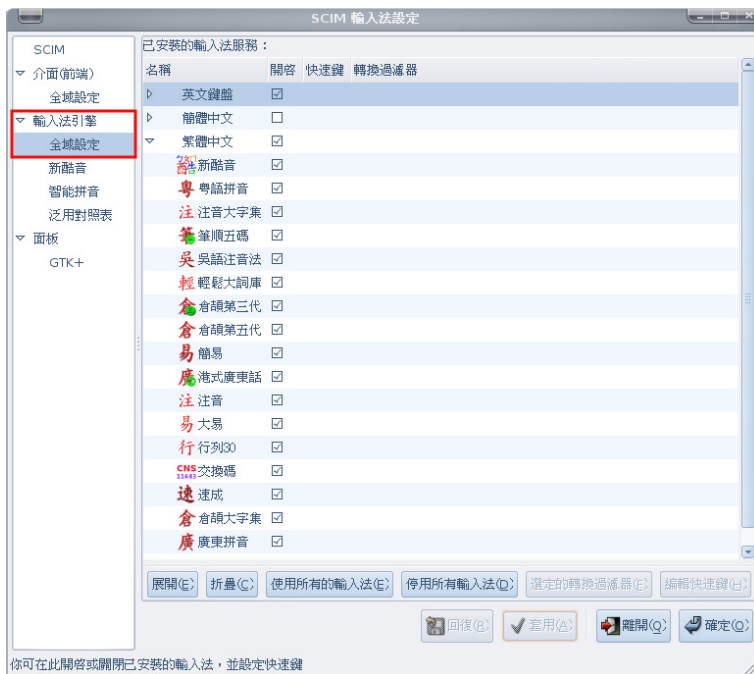


圖 7.73 選擇輸入法

7.4.3 gcin 中文輸入程式

1. 安裝 gcin

```
$ sudo apt-get install gcin
```

2. 切換 gcin 為預設的輸入法引擎

```
$ im-switch -s gcin
```

接下來只要登出登入就可以使用 gcin 囉！那麼我們來看看怎麼設定及使用吧！在 gcin 小圖示上按右鍵。

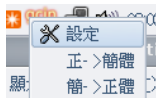


圖 7.74 選擇設定

接著會出現一堆 gcin 的設定選項，讀者可以自己玩看看。

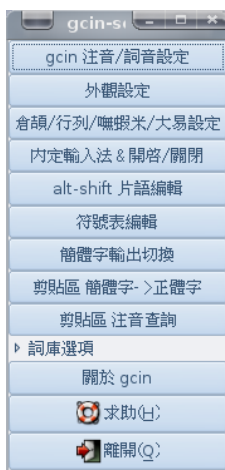


圖 7.75 gcin 設定選項

gcin 使用大易輸入法的情況



圖 7.76 gcin 大易輸入

gcin 的輸入法選擇，可以在輸入的時候點選**大易**，即可選擇別的輸入法，或是一般像 Windows 下那樣切換也可以，如果中文輸入出不來的話，要記得先開啟你要輸入的視窗，然後按 **Ctrl + Space** 即可。

```
內碼 ctrl-alt-0
倉頡 ctrl-alt-1
倉五 ctrl-alt-2
注音 ctrl-alt-3
標點倉頡 ctrl-alt-4
拼音 ctrl-alt-5
詞音 ctrl-alt-6
大易 ctrl-alt-7
行列 ctrl-alt-8
嘸蝦米 ctrl-alt-9
簡易 ctrl-alt--
日本式 ctrl-alt-=
greek ctrl-alt-`
```

圖 7.77 輸入法選擇

7.4.4 gcin 之嘸蝦米輸入法

首先呢！你要準備嘸蝦米的表格檔，因為嘸蝦米是商業軟體的關係，所以在這就不能說在哪下載囉！不過你還是可以用力的 google 一下，應該還是找得到，檔名為 liu5.cin，或是你在 Windows 裡所安裝的無蝦米資料夾裡可以找到。不過你下載回來的 liu5.cin 還要做一點小修改，像這個樣子。用 gedit 來開啟：

```
$ gedit liu5.cin
%gen_inp      #註：檔案的最前面
%ename liu5
%cname 無蝦米
%selkey 0123456789
%keyname begin
```

```

a A
b B
~中間略~
[ [
] [
%keyname end    #註：檔案前面的地方，往下找一點就可以看到。
%endkey
%chardef begin
a 對
aa 寸
~中間略~
]]v ㄣ
]]v ㄟ
%chardef end    #註：檔案最下方

```

這裡要注意！#的註解不可以打進去哦，從 windows 裡拿出來的 liu5.cin，前面有一堆 # 把它去掉，再來是在 %keyname end 的下面加上 %endkey，改這兩個地方看起來就和上面一樣了。那麼接下來要將 liu5.cin 這個檔案，轉換成為適用你系統的編碼，Ubuntu 是使用 UTF-8，但是你從網路上下載回來的很有可能是 big5，所以讀者改過 liu5.cin 後，可以用這個指令來轉檔。

```
$ iconv -f big5 -t utf-8 liu5.cin > liu5-utf8.cin
```

然後會產生一個 liu5-utf8.cin 的檔案，然後我們可以用 gcin2tab 這個指令，來將 liu5-utf8.cin 再轉成 gcin 可以用的 tab 檔案。

```
$ gcin2tab liu5-utf8.cin
```

接著把上個步驟產生的 liu5-utf8.gtab cp 到 /usr/share/gcin/table/ 裡。

```
$ sudo cp liu5-utf8.gtab /usr/share/gcin/table/
```

接著修改 gtab.list 這個檔案

```
$ sudo gedit gtab.list
```

改成像這個樣子：

```

內碼 0 -
倉頡 1 cj.gtab cj.png
倉五 2 cj5.gtab cj5.png

```

```

#輕鬆 2 ez7.gtab
注音 3 - juyin.png
#帶調粵拼 4 jyutping.gtab
#標點倉頡 4 cj-punc.gtab cj-punc.png
拼音 5 pinyin.gtab pinyin.png
詞音 6 - tsin.png
大易 7 dayi3.gtab dayi3.png
#行列 8 ar30.gtab ar30.png
嘸蝦米 9 liu5-utf8.gtab
#嘸蝦米 9 liu55.gtab
#嘸蝦米 9 liu57a.gtab
#嘸蝦米 9 noseeing.gtab noseeing.png
簡易 - simplex.gtab simplex.png
日本式 = kana-nippon.gtab kana-nippon.png
greek ` greek.gtab greek.png
#latin-letters ` latin-letters.gtab latin-letters.png

```

重點是嘸蝦米那幾行，只能有一個是沒有註解的（#字號就是註解），其他的都要用 # 字號註解起來。嘸蝦米後面的 9，表示使用 `Ctrl + Alt + 9` 就可以叫出嘸蝦米輸入法來使用，而 9 的後面是檔案名稱，要改成跟你剛才 `cp` 到 `/usr/share/gcin/table/` 的檔案名稱一樣。重新登入就可以用嘸蝦米輸入法囉！



圖 7.78 gcin 之嘸蝦米輸入法

如果你不是用 gcin 的話，請先用 `im-switch` 選擇後，再登入登入。

```
$ im-switch -s gcin
```

CHAPTER 8

網路應用程式

有人說：「Linux 沒有了網路就等於沒有用」，由此可知網路對 Linux 的重要性就跟我們每天都要喝水一樣。其實 Linux 不僅可以用來當 server 或是桌面環境使用，還有許多用途，像是嵌入式 Linux System，例如：Linux 手機、家電、點唱機、DVD 錄放設備等，都可以看得到。不過在這裡當然要介紹一下用於 Internet 上的應用程式，比方說在 Windows 上我們可以看到 IE、msn、Outlook、BT、Skype 等等的網路應用程式。

Linux 上也有許許多多像是這樣的應用程式，不過應用程式的名稱不一定和 Windows 一樣。Linux 的圖型介面其實是比 Windows 還要老的，其實在以前還沒有 Windows 的時候 Linux 就有圖型介面了，只是以前 Linux 的圖型介面較不完整，而且在那個時代電腦也不像現在這麼普及，加上 Linux 入門的門檻較高，所以以前的 Linux user 很少，更別說看到 Linux 的圖型介面了，所以 Linux 上有很多的應用程式都是 Linux 原創的，且都比 Windows 好用很多，尤其還有額外的特殊功能！不過當然也有在 Windows 看得到的軟體，同時也有適用於 Linux 的版本。那接下來我們就來看在 Linux 下有什麼常用的網路程式吧！

8.1 網頁瀏覽器 Firefox

hey！這個超有名的 web browser，連 Windows user 都在用。Firefox 有名是因為簡單好用，還有很多的擴充套件，當然在 Linux 上是一定有的囉！在 Ubuntu GNU/Linux 上預設是裝好的，所以在這裡就介紹一下 Firefox 的 Flash player、JAVA Runtime、Adobe Reader PDF、Mplayer plugin、Totem gstreamer plugin、Totem xine plugin 以及 Firefox 本身好用的外掛。

只要點左上方的地球圖示點選就可以執行了。



圖 8.1 Firefox 開啟 yahoo 首頁

Firefox 好用的書籤管理

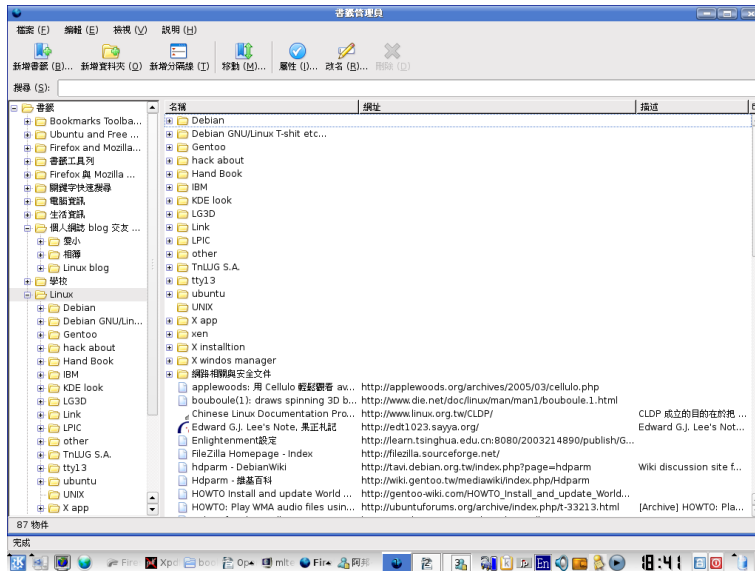


圖 8.2 Firefox Bookmark manager

假若你的 Firefox 是英文介面的話，可以透過以下指令來變成中文的呢！

```
sudo apt-get install mozilla-firefox-locale-zh-tw
```

8.1.1 Flash Player 安裝

在 Ubuntu 裡面的 multiverse 有包 Flash player，所以套件管理系統要把 multiverse 打開喔！在安裝的時候，它會問你要不要上網下載，就選 Yes 就好啦！

```
sudo apt-get install flashplugin-nonfree # 安裝 flash player
sudo update-flashplugin # 安裝到 firefox 上
```

接著開啟 yahoo 首頁看看 Flash Player 是不是裝好了。



圖 8.10 測試 Flash Player

是不是很简单呢？

8.1.2 JAVA Runtime Environment 安裝

現在，在 ubuntu 安裝 JAVA Runtime Environment 真的很簡單方便！從今年的 5 月 16 日後就不用到 sun 的網站下載 JDK 或 JRE 回來裝了，只要 apt-get 就可以啦！

```
sudo apt-get install sun-java5-jre sun-java5-plugin
```

這個動作可讓你下載 83MB 的檔案回來安裝，安裝的過程中會有授權條款的部份，只要選擇同意就可以繼續安裝程序，jre 和 java plugin 裝好就有 JAVA Runtime Environment 可以使用了，接下來我們到 java 官方的測試網頁測試看：

http://java.com/zh_TW/download/help/testvm.xml

安裝成功的話在網頁裡的**測試 JVM** 下面會有 JAVA 圖示跟你說你的 OS type、kernel version 以及 JAVA 的版本哦！



圖 8.11 JVM 測試網頁

8.1.3 Adobe Reader PDF 閱讀程式外掛

要裝 adobe reader 用 Firefox 的話，Ubuntu 已經幫你包進去了，然後我們只要再裝 cmap-adobe-cns1 可以讓 adobe 幫你載入中文字型。

安裝 mozilla 的 adobe reader 外掛，用下面這個 command 系統會自動地幫你把 adobe 的主程式套件 (acroread) 給安裝起來，所以會下載 55.8MB 的檔案：

```
$ sudo apt-get install mozilla-acroread
```

安裝好後，也不用重開 Firefox，它會自動幫你把 Adobe reader 外掛載入，第一次開啟 pdf 檔時，會先問你是否要同意授權，按下 Accpet。

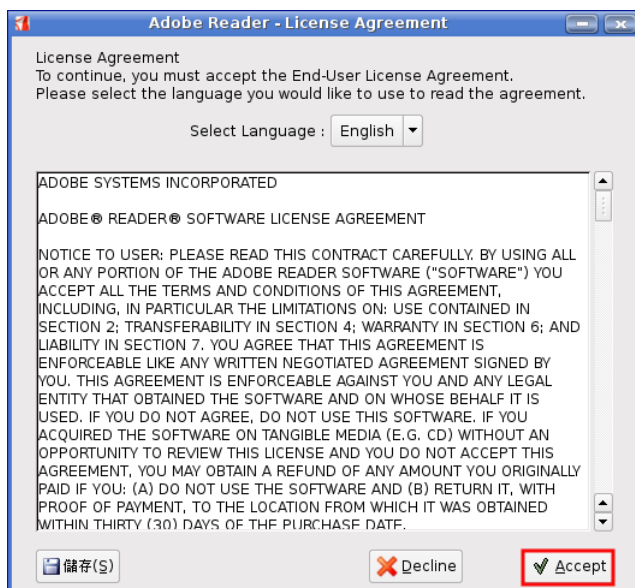


圖 8.12 是否同意 adobe license

接下來就會出現用 firefox 開啟 adobe reader plugin 讀取你的 pdf 檔囉！

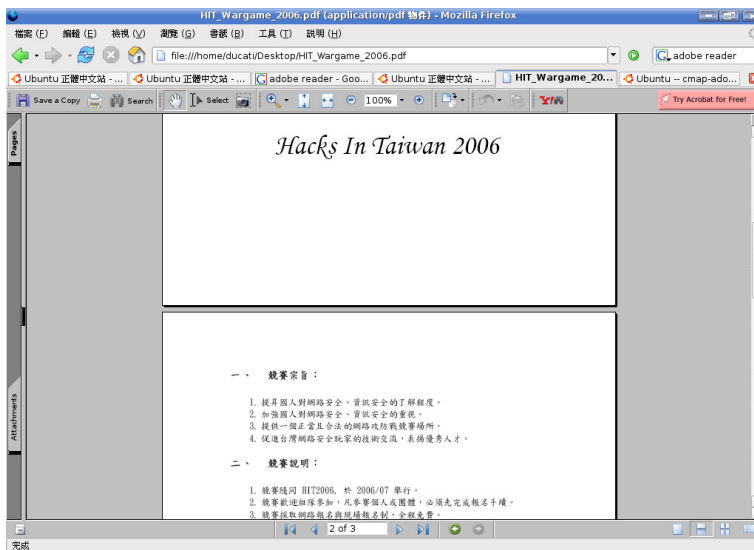


圖 8.13 用 Firefox 看 PDF

如果讀者不能順利的開啟 PDF 檔，有可能是少裝了 cmap-adobe-cns1，我們將它裝上去吧 ^ ^

```
$ sudo apt-get install cmap-adobe-cns1
```

8.1.4 Mplayer 多媒體播放器外掛

逛網的時候，常常會遇到有網頁內嵌影片，如果你以前是使用 Windows，IE 會自動幫你呼叫 Windows media player 出來幫你播放網頁裡的影片，所以今天我們要做的也是一樣的事情，我們可以用 apt-cache search "mplayer"，會發現有一個叫做 mozilla-mplayer - MPlayer-Plugin for Mozilla 的套件，還記得我們在前面有說過 firefox 是 based on Mozilla 的，所以裝上 mozilla-player 後 firefox 也可以用 Mplayer 來播放網頁中的影片哦！

🔧 安裝 mozilla-mplayer

```
$ sudo apt-get install mozilla-mplayer
```

這樣子就可以播放網頁中的影片囉！

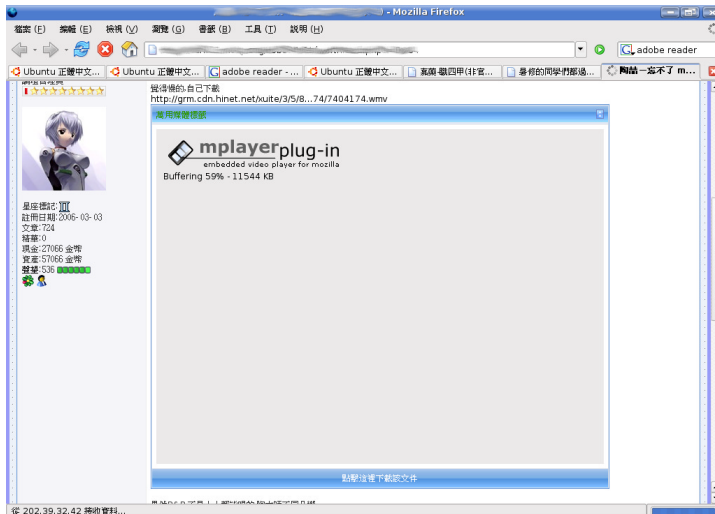


圖 8.14 影片讀取中



圖 8.15 影片播放中

8.1.5 Totem Gstreamer 多媒體播放器外掛

Firefox 的多媒體播放器有許多種 Totem Gstreamer 也是其中一個，可以依照需求或是喜好選擇一種來用即可，但是如果你有看過下一個多媒體章節就知道 Totem 的播放引擎同一時間只能用一個，如果你之前是裝 totem-xine 或 mozilla-mplayer 當 Firefox 的播放器，那麼安裝 totem-gstreamer-firefox-plugin 的時候，要先將 totem-xine-firefox-plugin 和 mozilla-mplayer 移除。

```
$ sudo apt-get remove totem-xine-firefox-plugin mozilla-mplayer
```

安裝 totem-gstreamer 和 totem-gstreamer-firefox-plugin

```
$ sudo apt-get install totem-gstreamer
```

那麼接下來把 totem-gstreamer-firefox-plugin 安裝上去

```
$ sudo apt-get install totem-gstreamer-firefox-plugin
```

再來一樣開啟有內嵌影片的網頁試試，雖然播放介面還是跟 Totem-xine-firefox-plugin 一樣是 Totem，只不過是播放引擎不一樣。



圖 8.16 Totem Gstreamer 多媒體播放外掛

8.1.6 Totem xine 多媒體播放外掛

要使用 Totem-xine-firefox-plugin 當你的 Firefox Multimedia plugging 前，你必需要先將 totem-gstreamer-firefox-plugin 和 mozilla-mplayer 移除掉，不然你裝了 Totem-xine-firefox-plugin 後，Firefox 在播放網頁上的影片時，還是一樣會用 Mplayer 或 Totem-gstaram-firefox-plugin 來播放。

✎ 移除 Totem-gstaram-firefox-plugin 和 mozilla-mplayer

```
$ sudo apt-get remove totem-gstreamer-firefox-plugin mozilla-mplayer
```

✎ 安裝 Totem-xine-firefox-plugin

```
$ sudo apt-get install totem-xine-firefox-plugin
```

接下來開啟一個有內嵌影片的網頁吧，這種網頁通常可能會在討論區看到，要請讀者自己找看看，在這裡筆者用班網上的來做示範圖片。



圖 8.17 Totem xine 多媒體播放外掛

8.1.7 一些好用的外掛

相信很多人都用過 Mozilla Firefox 後，都覺得它那可以更換的佈景主題和很多好用的擴充套件很棒，那麼接下來我們就來看有什麼實用的「擴充套件」吧！

1. 同步你不同電腦上的 Bookmarks：Bookmarks Synchronizer 3

這是個超方便的書籤。同步軟體的原理是將你的 bookmarks 上傳到你的網路空間，其可以適用的傳輸通訊協定有 FTP、HTTP、HTTPS，所以可以利用你已有的網路空間，例如：ISP 廠商提供給你的或是學校的網頁空間，以下我們用 FTP 來做為示範

安裝 Bookmarks Synchronizer 3

網址 <https://addons.mozilla.org/firefox/1989/> 點選 **install now**，或是到官方網站 search 一下 bookmarks synchronizer

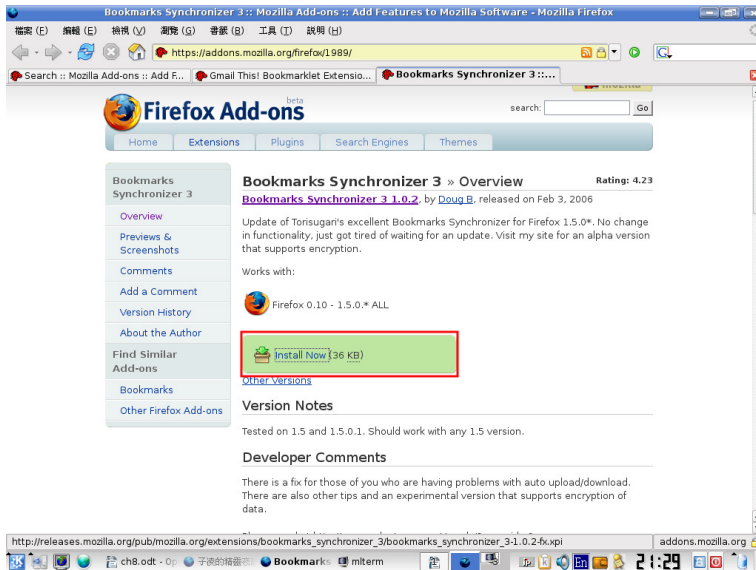


圖 8.18 Bookmarks Synchronizer 3 頁面

下載完後會出現一個小視窗問你是否要安裝 Bookmarks Synchronizer 請點選
安裝



圖 8.19 安裝 bookmarks synchronizer

按下來 Firefox 會要求你重新啟動 Firefox 來完成 Bookmarks Synchronizer 安裝



圖 8.20 重新啟動 Firefox

✎ 設定 bookmarks synchronizer 3

接下來重新啟動 Firefox 後請點選 工具--> 擴充套件

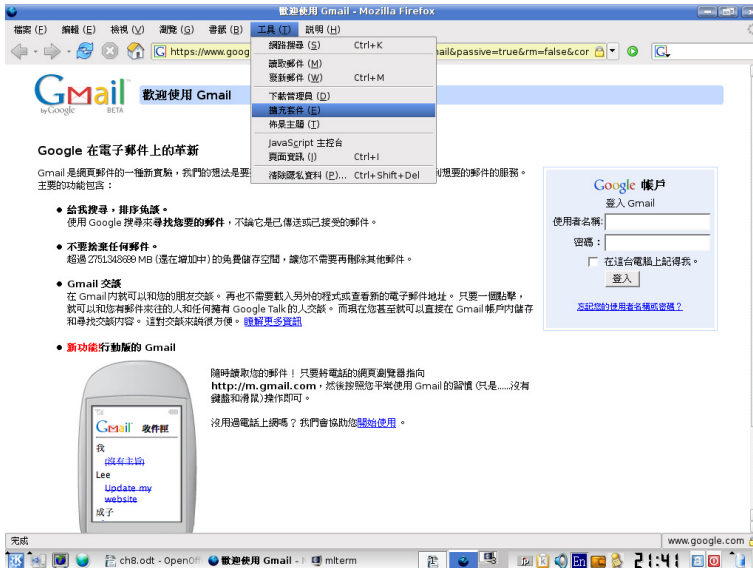


圖 8.21 擴充套件選項

然後選擇 bookmarks synchronizer 3 --> 偏好設定，來設定你的主機名稱和帳號密碼。

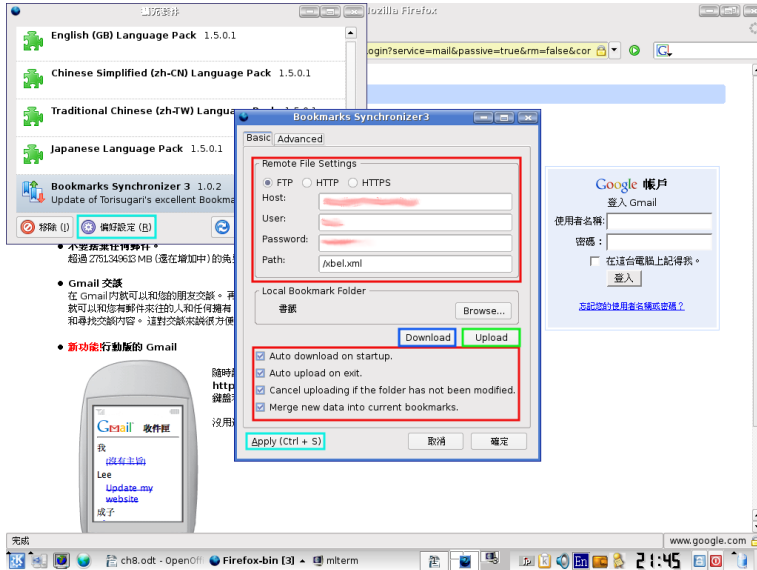


圖 8.22 設定你的 host name 和帳號密碼

選擇 FTP，接著輸入你的主機名稱、帳號、密碼，然後再將下面的四個選項打勾，都打勾的話，會在你開啟 Firefox 自動下載 Bookmark，以後關閉 Firefox 時會上傳 Bookmark。設定好了要按一下 Apply (Ctrl + S)，接下來可以按下 Upload 將你的 Bookmark 上傳到你的 FTP 主機去。

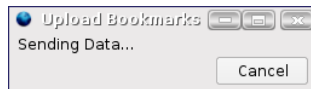


圖 8.23 Bookmark 上傳中

然後另外一台要同步 Bookmarks 的電腦也是照一樣的方法將 bookmarks synchronizer 3 安裝上去即可，只要按下 Download 就可以同步你的 Bookmarks 了。

2.來上 IRC 聊天去：ChatZilla

嘿嘿！用 Firefox 也可以上 IRC 聊天，很方便吧！讓你不用再多開一個程式就可以上 IRC 聊天！

安裝 Chatzilla

網址 <https://addons.mozilla.org/firefox/16/> 一樣點選 install now

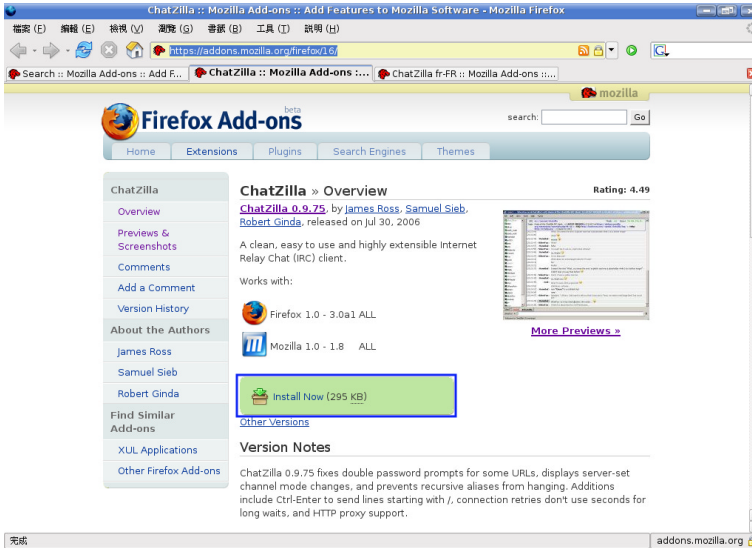


圖 8.24 Chatzilla 安裝網頁

點選後一樣點選 立刻安裝



圖 8.25 安裝 Chatzilla

再來就重新啟動 Firefox



圖 8.26 重新啟動 Firefox 完成 chatzilla 安裝

接著可以在網址列上打上你想連上的 irc server 例如：irc://irc.freenode.net

或著也可以點選 工具--> chatzilla 來啟動

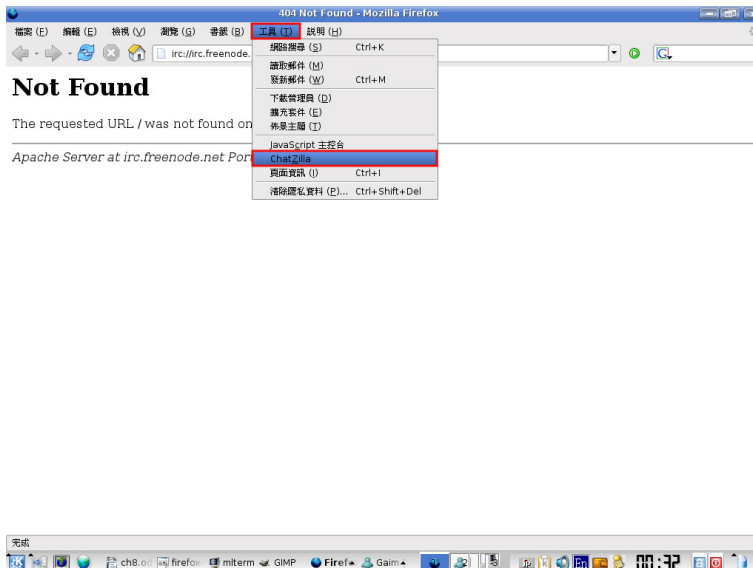


圖 8.27 啟動 Chatzilla

連上 IRC server 後可以在下面的對話框輸入你想加入聊天的 channel 例：

/j #ubuntu-tw

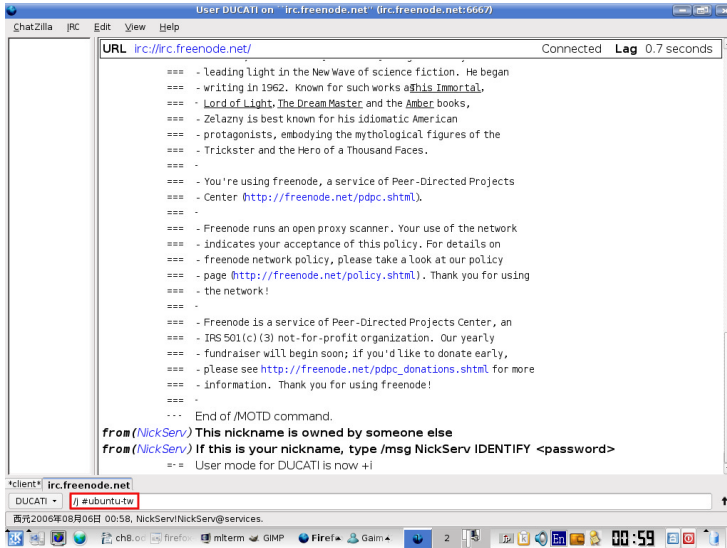


圖 8.28 連上 ubuntu-tw

連上去啦！是不是很方便呢？



圖 8.29 Chatzilla 聊天畫面

✎ 設定 Chatzilla

那麼要是我們連上去的 channel 是使用 big5 編碼的話要怎麼設定呢？簡單的說明一下，點選 Chatzilla 右上方的 Chatzilla--> Preferences

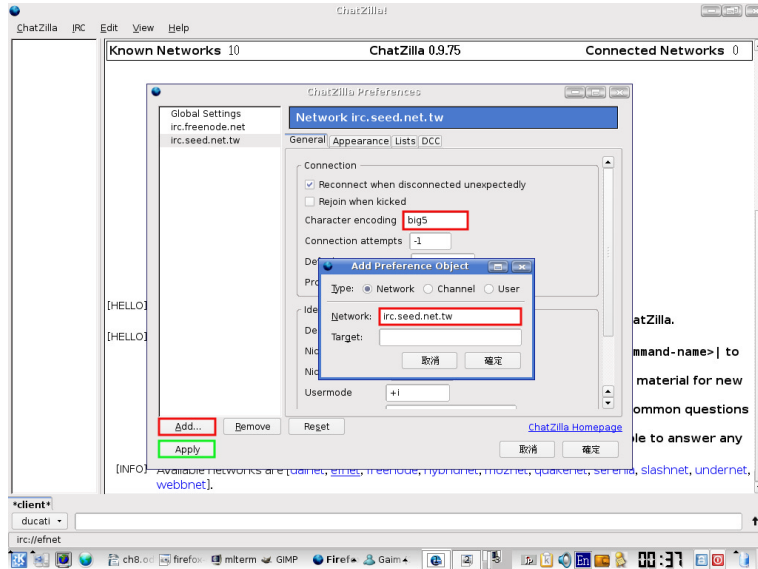


圖 8.30 設定編碼

在你加入過的 server 或是 channel 選擇 General，然後在 character encoding 輸入 big5 即可。那麼如果我們想讓 Chatzilla 開啟後、自動載入我們想上的 IRC server 的話，可以這樣子做：一開始一樣把 Preference 叫出來後選 Global Settings--> Startup，然後下面有個 Add icon 按下後，輸入你想要 startup 的 server 就可以了。

8.2 E-Mail 收信軟體

Linux 上是否有像 Outlook 一樣的收信軟體呢？答案是有的，接下來會介紹兩個 E-Mail 收信軟體，Thunderbird 和 Evolution 其實操作設定上都和 Outlook 差不多，可以設定是否加密、信件分組、是否保留信件在原本的信箱裡，接下來讓我們來看看基本的帳號設定。

8.2.1 Thunderbird (包含 RSS feed)

Thunderbird 是 Mozilla 同系列的軟體，不過 Thunderbird 目前只有簡體中文的版本（之所以安裝 zh-cn 翻譯檔的原因，是因為官方沒有收錄 zh-tw 的翻譯），Thunderbird 的特色包括：靈活的垃圾郵件控制、RSS 閱讀、可以下載你所有的 E-Mail、保存搜尋你的 Mail、信件分組、隱私保護，在這裡我們就以 Gmail 為例介紹一下如何用 Thunderbird 來收 Gmail 的信。

啟動 Gmail POP 功能

```
http://mail.google.com/support/bin/answer.py?answer=13273
```

安裝 Thunderbird

```
$ sudo apt-get install thunderbird-locale-zh-cn
```

裝好了以後可以在 應用程式--> 網際網路--> Thunderbird 找到並啟動，一開始會問你要用什麼方式來建立一個帳號來使用 Thunderbird，我們選第一個 **電子郵件帳號** 按下一步。

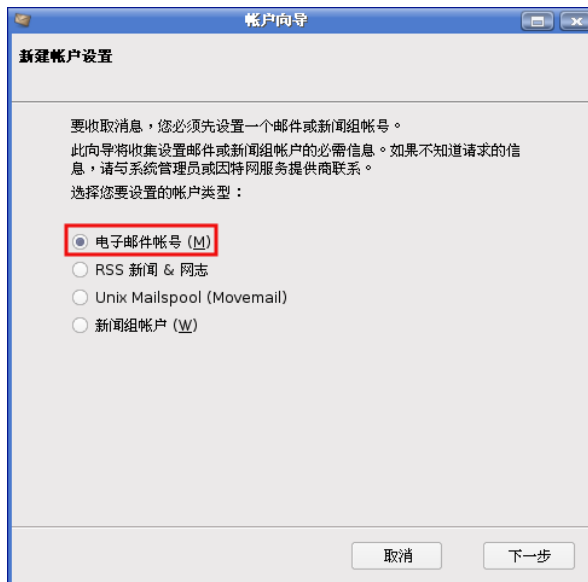


圖 8.31 用已有的 E-Mail 開啟

接下來輸入你的名字和帳號名稱，接著按下一步。

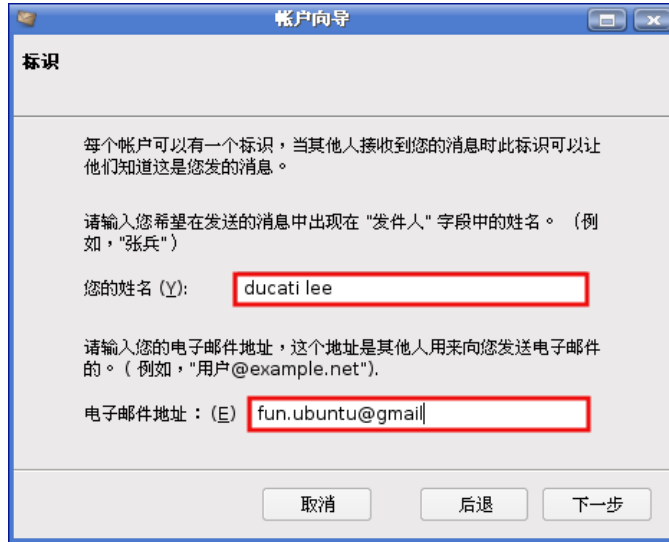


圖 8.32 輸入名字帳號

再來輸入 pop3 和 smtp 伺服器網址（用來收信和發信用的），接著按下一步。

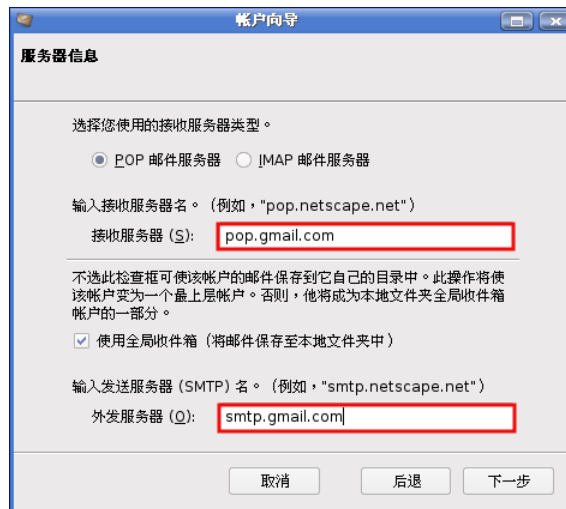


圖 8.33 輸入 pop3 和 smtp

然後輸入你在 Gmail 的帳號也包括“@gmail.com”，通常都是一樣的。



圖 8.34 輸入你的帳號

接著輸入你的帳號資訊，這裡看愛打什麼都可以。

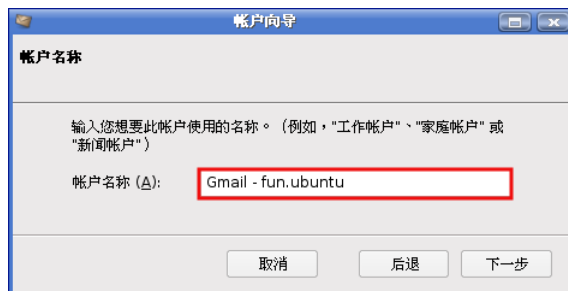


圖 8.35 帳號資訊

驗證一下跟剛才輸入的資訊是不是一樣，如果沒問題就可以按完成囉！



圖 8.36 驗證帳號資訊

不過這樣子還是不能收信，我們還要設定一下 port 和連線方式。請依下圖將 port 改成 995 (不過不是每個 E-Mail server 都是一樣的設定，讀者需要自己查一下你的 Mail server)。先將帳號設置的視窗叫出來：編輯--> 郵件/新聞帳戶設置。

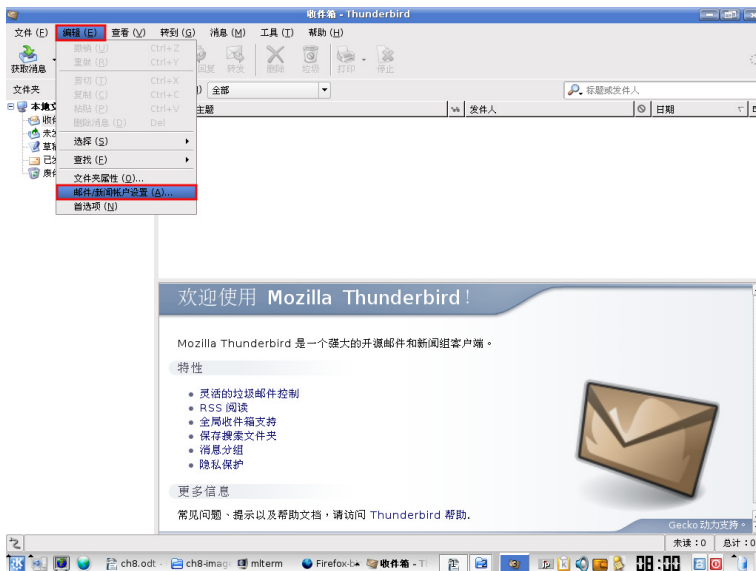


圖 8.37 帳號設置

將 port 改成 995、連線方式 ssl。

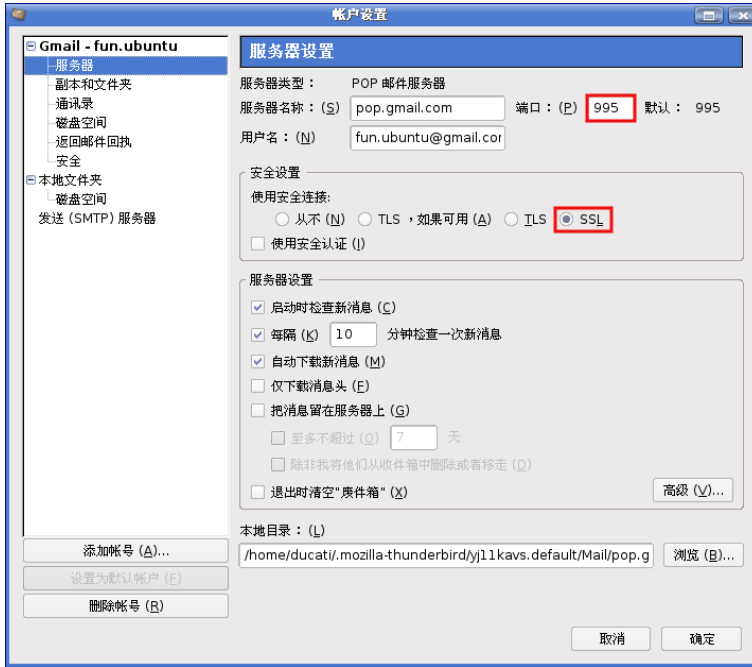


圖 8.38 改 port 與連線方式

這樣就可以收信囉！你也可以多設幾個帳號，這樣子就不用每次都要跑不同網頁去收信。

8.3 即時通訊軟體

在這個網路資訊發達的時代，幾乎是人人都會使用網路即時通訊軟體來聯絡彼此。時下，在 Windows 環境下最流行的就是 MSN 和 yahoo 推出的即時通。雖然 yahoo 曾推出或 Linux 版本的即時通，不過裝了以後覺得不是很順手，後來又重回 gaim 的懷抱。那 Linux user 都是用什麼樣的軟體連上 msn 和即時通呢？最常見的一定是 Gaim 了，而且 Gaim 支援的通訊協定不是只有 msn 和即時通，還有 IRC、ICQ、GroupWise、Jabber（可以用來 Google Talk）、Napster 這麼多！看得眼花撩亂了，還有 aMSN 也很常見喔！不過 aMSN 只能使用 MSN 的通訊協定，接下來請看筆者我娓娓道來吧！

8.3.1 Gaim 支援最多通訊協定的即時通訊軟體

Gaim 是一隻很特別的通訊軟體，支援相當多的通訊協定，包括有 AIM/ICQ、GroupWise、IRC、Jabber、MSN、SIMPLE、Yahoo、Zephyr，相當多吧！接下來筆者將介紹一下大概的功能以及使用方式等等，還有一些特別的功用。

建立帳號

因為 Gaim 是一個將很多協定統一的一個前端介面，所以我們不必太擔心在建立帳號時會有繁雜的選項，反之只要在建立時選擇我們要的通訊協定。在選擇通訊協定時，需視你協定所需的資訊而決定。

接下來開啟 Gaim，應用程式--> 網際網路--> Gaim 網路即時通，點選帳號清單--> 新增，就可以選擇你想要的通訊協定，如果沒有的話也可以在主程式裡的帳號--> 新增/編輯 找到。接下來按**新增**來加帳號吧！



圖 8.47 Gaim 新增帳號

在**登入**選項的地方先選擇一下**通訊協定**，假設這次我們想新增的協定是 MSN，就在這裡選 MSN



圖 8.48 使用 MSN 通訊協定

填好你先前已有的 MSN 帳號密碼，也可以使用好友圖示哦！



圖 8.49 填好你的帳號密碼

假如想要增加 Yahoo 帳號，也差不多如下圖。

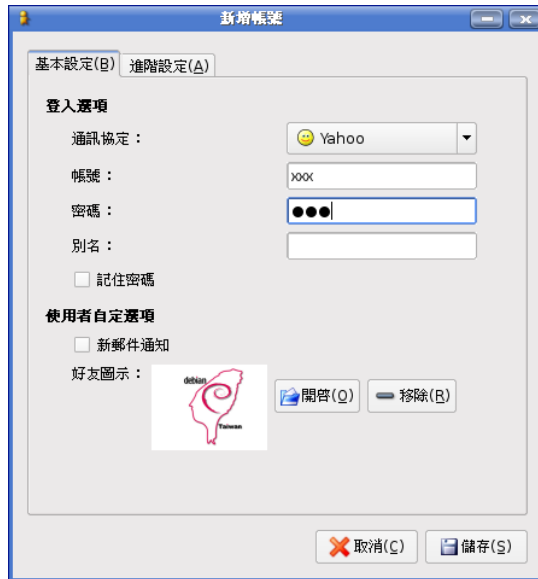


圖 8.50 新增的 yahoo 帳號

新增好了以後就可以在**帳號清單**裡看到了



圖 8.51 新增好後會在帳號清單裡出現

讀者可以看到在清單裡多了剛才新加的 MSN 帳號，共有三項資訊：帳號、啟動和通訊協定，只要在你想上線的帳號旁的**啟動**打勾就可以上線囉！讀者也可以增加兩個不同的協定，好友清單和聊天視窗都是整合在一塊的，蠻方便的感覺。



圖 8.52 整合式的好友清單

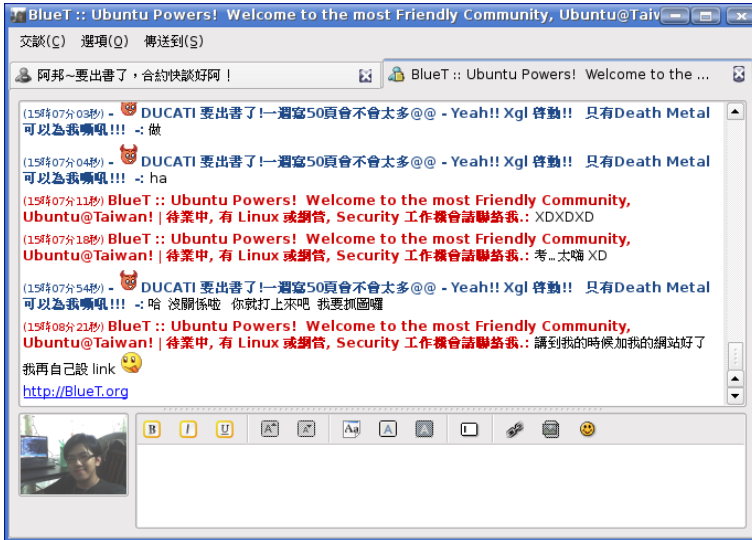


圖 8.53 整合式的聊天清單

其實 Gaim 的功能不只如此，它還有以下優點：

1. 用 Gaim 登入 MSN 時，別人把你的視窗關了也會有即時訊息在聊天視窗中出現。

備註 舊版跟新版 2.0 出現的訊息不一樣，不過訊息因為不是很準確，只能當參考用。

2. 當你上線時，Gaim 可以先提取你好友的照片方便閱覽。而且假如你是第一次使用 Gaim 你或許會看到：哦！這個人不是沒上線嗎？可是 Gaim 確可以提取到他的好友圖示。那就表示你的好友是隱身上線的囉！
3. Gaim 可以隱身聊天！嘿嘿！這功能雖然沒有什麼，不過有時候是滿好用的，讀者可以試試跟朋友聊天時，將 Gaim 的 MSN 狀態變成隱身會發生什麼事呢？不過前提是要先與朋友聊了天並且對方有回覆你才可以。
4. Gaim 可以讓你知你的好友是不是將你的帳號從他的好友清單中刪除了！這個功能可以直接在你的好友清單中發現，只要你看到好友清單上的好友圖示是打上一個紅色的 X，此即是你的帳號在你好友的清單中給刪除了。如果是這樣，快點去找你的好朋友聊一聊吧！不然別人會以為你忘了他哦！
5. Gaim 可以透過除錯視窗查驗你的好友是不是將你封鎖？筆者也是在網路上閒逛時看到舊文章而發現的，不過筆者並沒有試過，有興趣的讀者可以試看看哦！
6. 不用裝外掛就可以自行設定你的好友暱稱。
7. 捕捉好友狀態！例如：好友的登出、登入、離開、閒置，也可以設定某個好友或是多個好友上線時，自動傳訊給她。
8. 取得你的好友資訊！在好友清單上按右鍵可見。點選後可以發現你的好友曾在 MSN 官方網站上曾經打過的自我介紹。

不過再完美的軟體還是有一些缺陷：

1. 不能看到好友 MSN 的第二個狀態。
2. 好友的 MSN Space 不能在清單中點選。

3. yahoo 的好友狀態因為編號的關係顯示出來是亂碼。
4. 不能使用視訊和語音聊天。

8.3.2 skype 目前網路電話的領先者

在這網路資訊發達的時代，自然有在網路上替代一般電話的通訊方式，就是網路電話。而 skype 是目前全世界最多人用的網路電話，然而會造成這樣的情況也是因為它簡單的介面，除了可以用語音聊天之外，還可以用影像通話！並且在各個常見的 OS（作業系統）上都可以看得到 skype 的蹤影，在 Linux、Windows 以及蘋果電腦的作業系統上都看得到。skype 應該是會上網的人大概都會用了，所以在這裡就簡單介紹一下功能就好，重點放在怎麼在 Linux 上使用。

安裝 skype

由於 Skype 在 Ubuntu 裡面還沒有，所以要自己在 sources.list 裡加上底下這行套件源，然後再 apt-get update 吧！

```
deb http://download.skype.com/linux/repos/debian/ stable non-free
```

開始安裝啦！

```
$ sudo apt-get install skype
```

不過，如果你要在 Skype 裡輸入中文，必需要再裝上 qt-immodule，而且這裡需注意的是你裝的輸入法引擎 qt-immodule 也不一樣，scim 的是 scim-qtimm 可以用 apt-cache search 一下。

```
ducati@X20:~\$ sudo apt-cache search qt-immodule
scim-qtimm - SCIM context plugin for qt-immodule
```

安裝 scim 的 qt-immodules

```
$ sudo apt-get install scim-qtimm
```

search gcin 的 qt-immodules 先 search 一下，跟剛才的 search scim 的差不多。

 search 的結果：

```
gcin-qt3-immodule - QT input method module with gcin as backend
gcin - a input method platform, supports GTK/QT immodule and XIM
gcin-table-sakura - Big-5 Nippon Kana IM table for gcin
gcin-tables-cn - Simplified Chinese IM tables for gcin
gcin-tables-kr - Korean IM tables for gcin
gcin-table-boshiamy - a installer for Boshiamy input method of gcin
```

可以發現名稱不一樣。

 安裝 gcin-qt3-immodule

```
$ sudo apt-get install gcin-qt3-immodule
```

接下來開啟 Skype 來試試，應用程式--> 網際網路--> Skype。

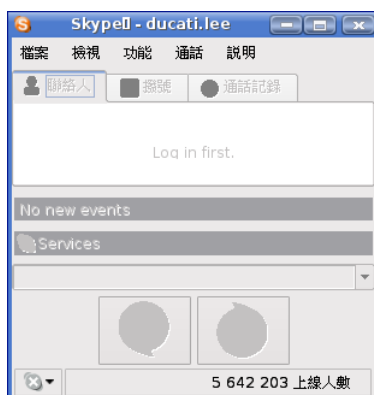


圖 8.64 Skype

用原本已有或是新的帳號登入，檔案--> 新身份登入

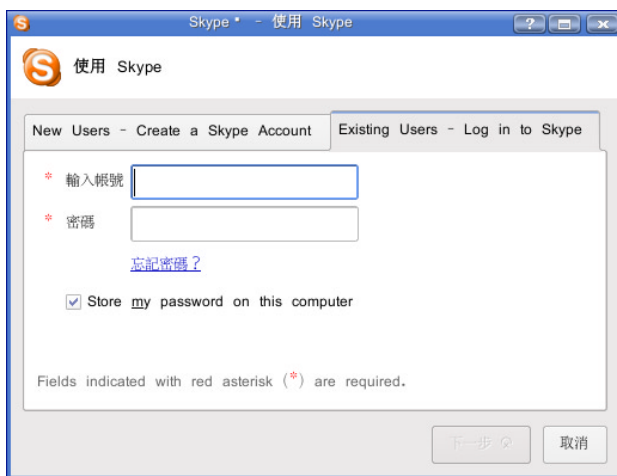


圖 8.65 已有帳號登入



圖 8.66 申請新的帳號

申請新的帳號在這輸入



圖 8.67 new account

再來就是登入後的畫面

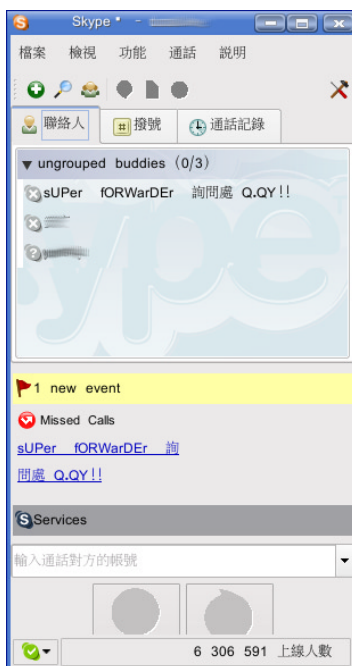


圖 8.68 登入後畫面

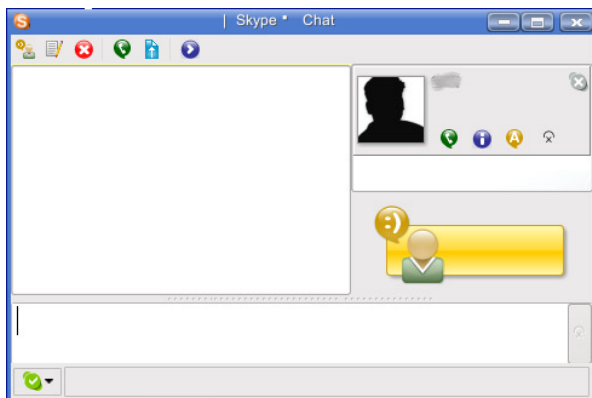


圖 8.69 對話框



圖 8.70 播出



圖 8.71 通話記錄

8.3.3 Ekiga 支援 SIP 協定的網路視訊電話

Ekiga 是 ubuntu 內建的網路視訊電話，免費不用錢而且安裝簡單，註冊時也不用輸入你的詳細資料，只要 email 和你的 ID 就可以了。應用程式--> 網際網路 --> Ekiga，第一次啟動後會一步一步的引導安裝註冊。

安裝 Ekiga

```
$ sudo apt-get install ekiga
```

開始第一次設定

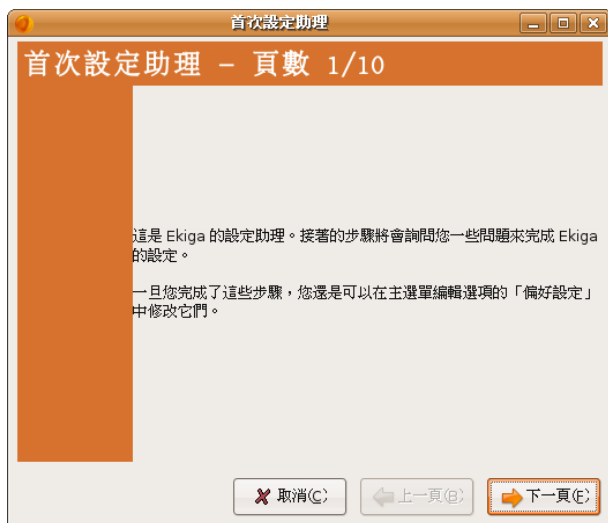


圖 8.72 首次設定助理

輸入你要顯示給別人看的名字



圖 8.73 輸入姓名

註冊並取得你的 ekiga.net SIP 免費帳號

圖 8.74 開始註冊

點選 subscribe 註冊

圖 8.75 註冊帳號

輸入你的姓名、email、時區還有帳號名稱以及密碼

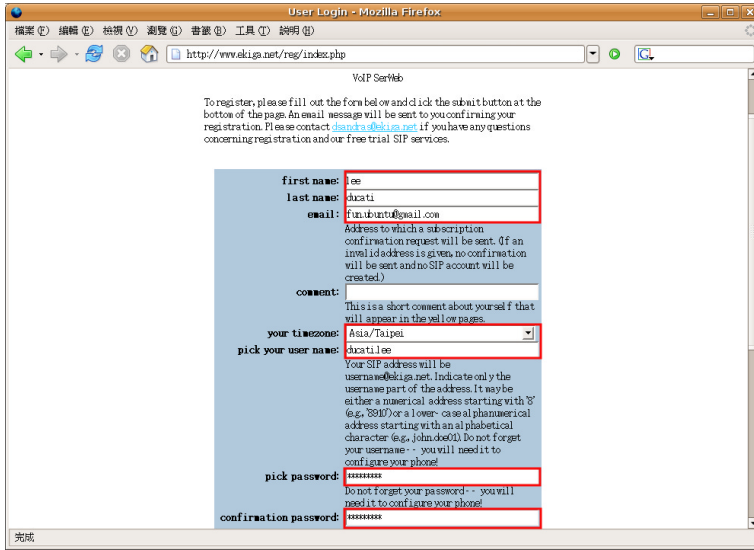


圖 8.76 輸入註冊資訊

同意條款的部份，將 I Accept 打勾，然後點選 Register

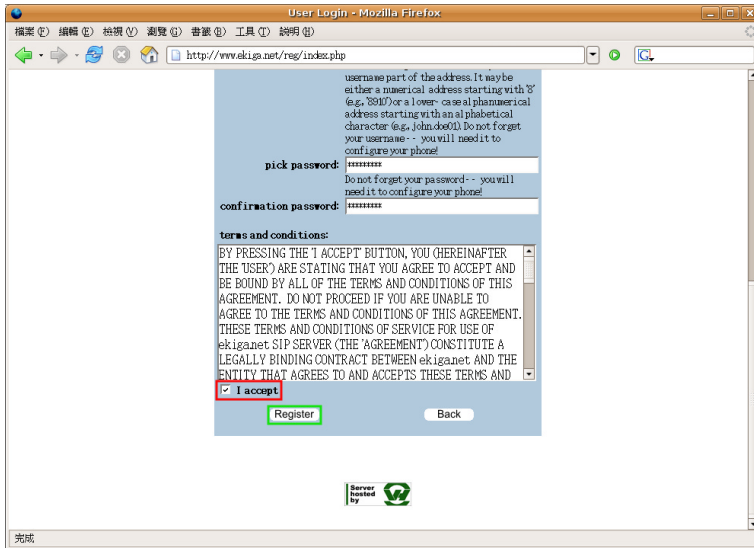


圖 8.77 是否同意條款

接著會出現一個謝謝你安裝的網頁

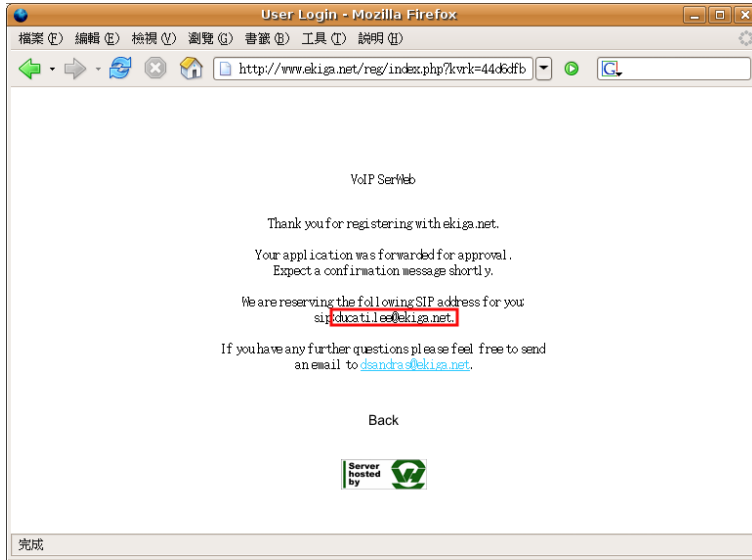


圖 8.78 註冊完成

然後到你剛剛註冊時輸入的 email 的地方啟動你的帳號，可以在 24 小時內啟動帳號，點最下面最長的那個網址。

Your ekiga.net Registration 收件匣

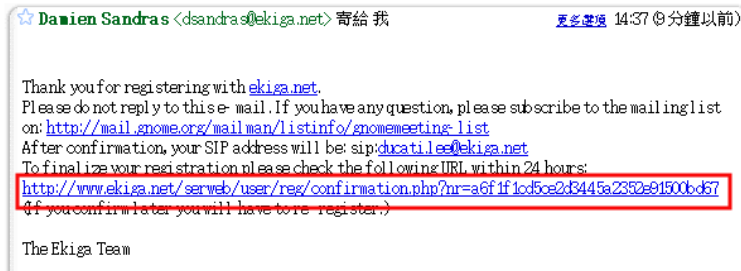


圖 8.79 啟動帳號

啟動完成後會出現一個恭喜你完成啟動的網頁

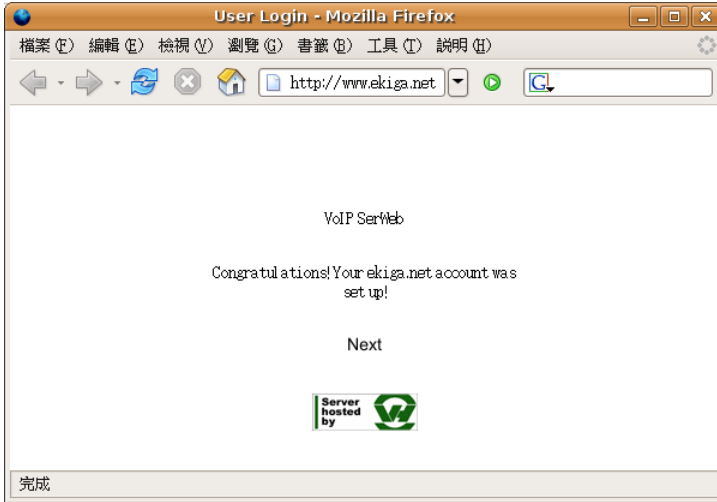


圖 8.80 完成啟動帳號

接著我們再回到設定助理，輸入你的帳號密碼，按下一頁



圖 8.81 輸入帳號密碼

選擇網路連線的類型



圖 8.82 連線類型

然後 Ekiga 會自動幫你偵測 NAT



圖 8.83 NAT 類型

選擇音效管理程式，ubuntu 預設是 ALSA，所以我們用 ALSA 就可以了



圖 8.84 音訊管理員

選擇音訊裝置，主要是音訊輸出設成 Default 就可以了



圖 8.85 音訊裝置

設成功後按**測試**，沒問題的話可以聽到自己的聲音，不過麥克風聲音要先調大，否則聲音放出來會很小。



圖 8.86 錄放音測試

Ekiga 也可以用影像對話，所以選擇是你的視訊管理程式，一般狀況用 V4L 就可以了。



圖 8.87 視訊管理員

如果你沒裝視訊，Ekiga 將找不到裝置



圖 8.88 視訊管裝置

終於設定完成啦！按套用就可以開始聊天了。



圖 8.89 設定完成

Ekiga 的使用者介面，只要在長方形的框框輸入你好友的帳號，按下右邊小插頭的圖示即可撥號給你的好友。



圖 8.90 Ekiga

8.4 網路聊天程式

現在網路連線的普及甚至是比電話費還便宜了，因為網路費大多都是月固定制，所以有很多網路通訊軟體也很普及了。在 Linux 的世界中 IRC 是不可或缺的，事實上很多 Linux 社群、寫程式的社群、自由軟體社群、甚至是很多同好會，也都會在 IRC channel 閒聊打屁或是聊正事、請教問題都會用 IRC 來做溝通的管道，另外當然還有 BBS 啦！雖然筆者並不是很常上 BBS，但是今天一定要介紹一下 PCMan 先生寫的 PCManX，接下來我們就來看看怎麼用 Xchat 來連上 ubuntu-tw，還有用 PCManX 上 BBS。

8.4.1 PCManX—自由的 BBS 連線軟體

PCManX 是個很有名的 BBS 連線軟體，在 Windows 上也流傳許久，連筆者不是用 Linux 的好友也在用。方便的介面以及可以自動幫你轉 big5 編碼，在 Linux 上用 PCManX 來連 BBS 是再好不過的選擇。

✎ 安裝 PCManX

```
$ sudo apt-get install pcmanx-gtk2
```

執行 PCManX，應用程式--> 網際網路--> PCManX

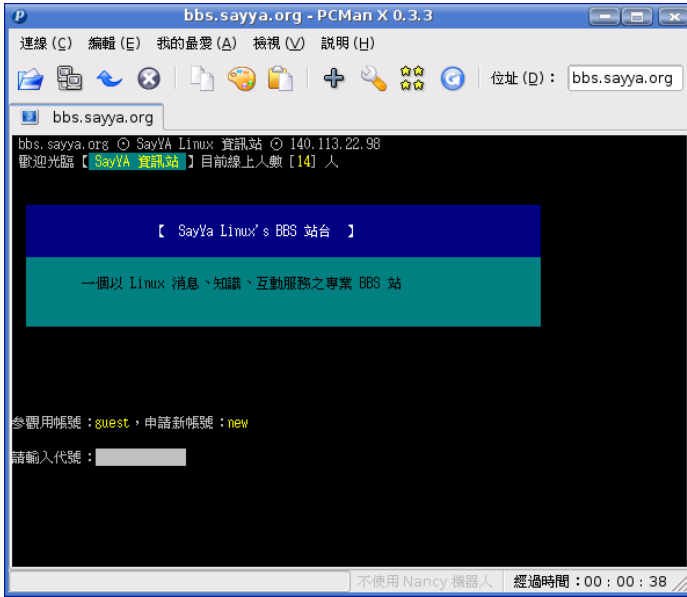


圖 8.91 PCManX

還有很多功能哦！如，自動回覆機器人、我的最愛、字型設定等等。而且還有防閒置功能讓你不會被伺服器斷線。

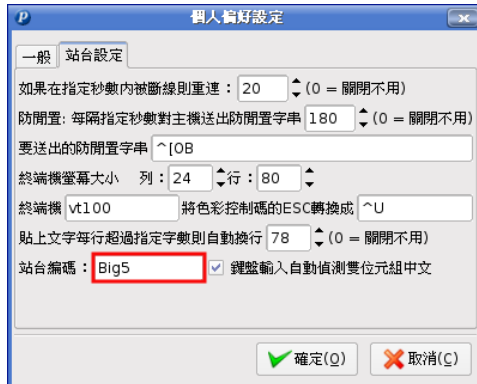


圖 8.92 站台設定



圖 8.x 一設定

也有很方便的站台列表供你尋找使用

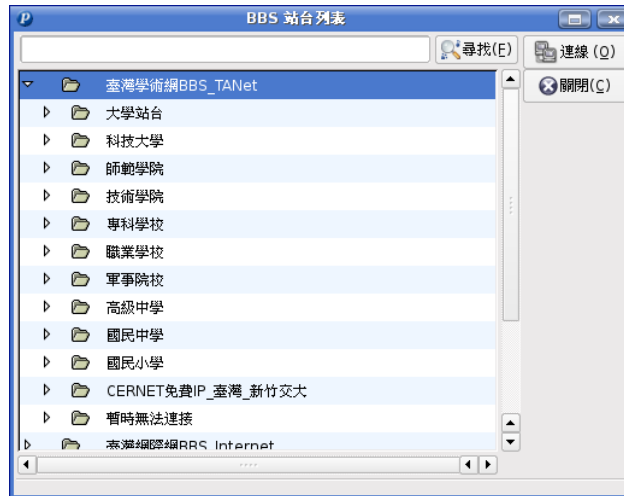


圖 8.93 站台列表

8.4.2 Xchat—可連上高手眾多的 IRC 聊天室

Xchat 也一直是 Linux 上有名的聊天軟體，上面有很多的 Linux user group 以及自由軟體的社群和程式，IRC 一直是有許多高手聚集的地方，不管是上去打屁聊天或是問問題都是個很好交流彼此意見的地方，以下示範怎麼用 Xchat 來連上 #ubuntu-tw。

✎ 安裝 Xchat

```
$ sudo apt-get install xchat
```

✎ 啟動 Xchat

應用程式--> 網際網路--> Xchat，開啟後會有一個列表和設定你暱稱的地方，Linux 的社群大部份都是在 FreeNode 上，ubuntu 也是。



圖 8.94 設定暱稱和選擇 IRC server

設定好後、按**連接**，接著會問你想要加入哪個 channel，這個時候我們可以輸入 #ubuntu-tw 哦。

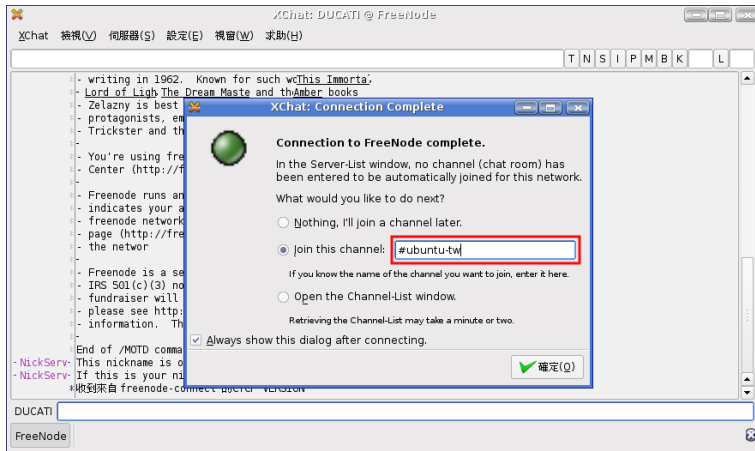


圖 8.95 加入 ubuntu-tw

接下來就可以跟眾多高手聊天打屁啦！

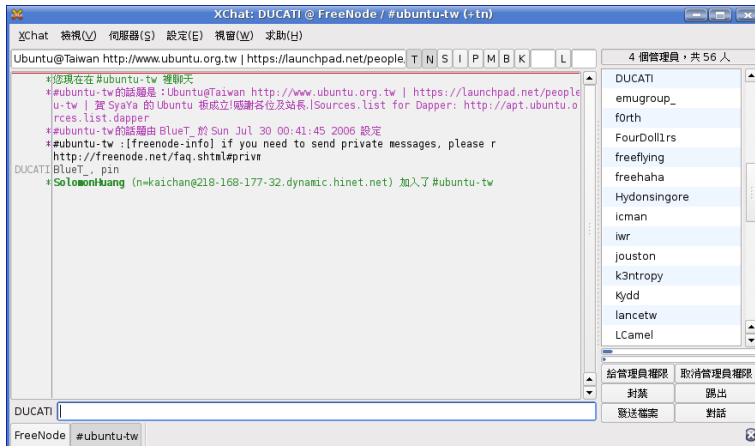


圖 8.96 #ubuntu-tw channel

我們也可以到 Tossug 的 channel 去逛逛，將 Tab 切換到 FreeNode 後輸入 `/j #ubuntu-tw`，這個指令的意思是說 join ubuntu channel。

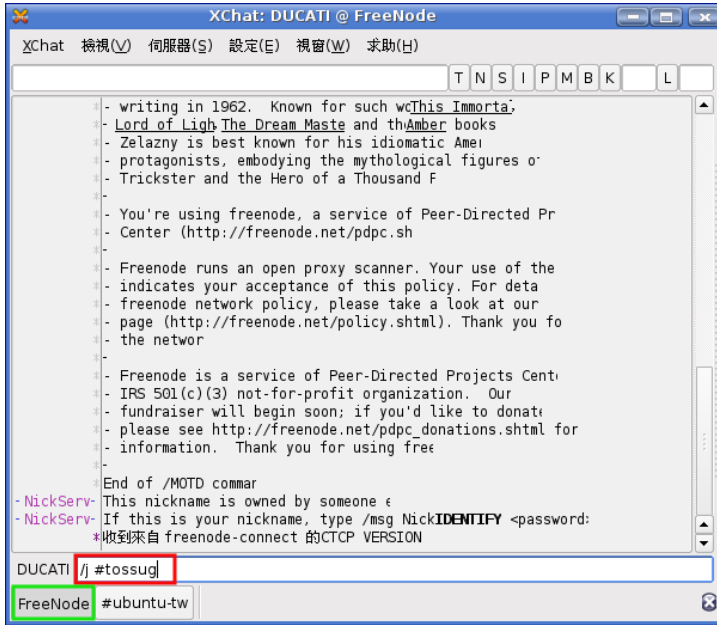


圖 8.97 加入 tossug channel

在 tossug channel 裡聊天

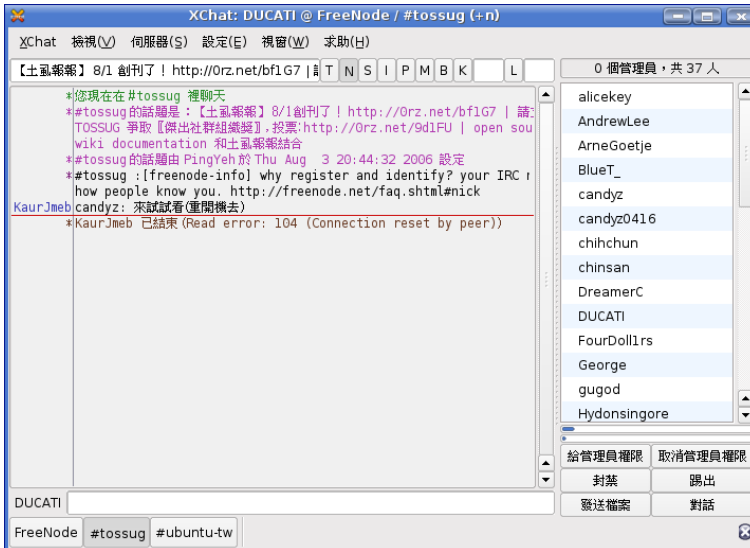


圖 8.98 #tossug channel

8.5 檔案傳輸軟體

檔案傳輸是一定要的啦！不知道你是否偷偷用學校的頻寬來分享檔案的經驗呢？或者是朋友架給你用的呢？關於這個，筆者我倒是有時候會用來下載影片或是音樂什麼的。我們來看看兩個 FTP client，一個是 gFTP 老牌的 FTP client，想當初剛玩 Linux 的時候我也都是用這個，再來是 Filezilla 新一代的 FTP client，為了走向國際化，現在大部份的 Linux 預設編碼都是用 UTF-8，不過要是連上的 FTP 站台若是 Big5 呢？那用 Filezilla 就對啦！好看好用又方便。

8.5.1 Linux 上老牌的 FTP 軟體－gFTP

✎ 安裝 gFTP

```
$ sudo apt-get install gftp
```

這樣就好了，其他有三個相依的軟體會自動安裝，所以總共有四個軟體一起安裝。

接下來從選單叫出來，應用程式--> 網際網路--> gFTP

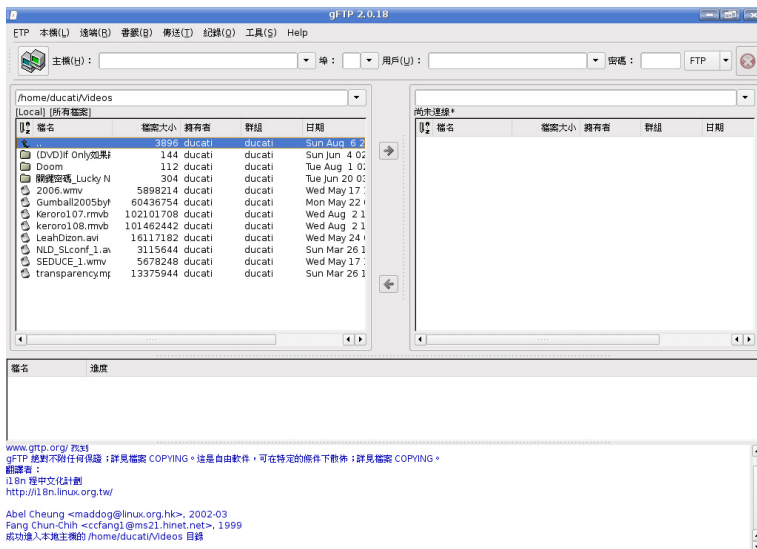


圖 8.99 gFTP

內建的 Linux 站台

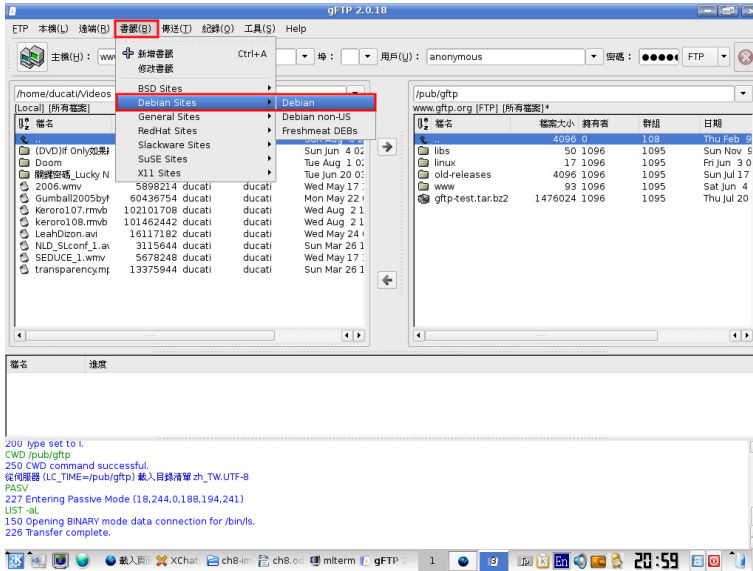


圖 8.100 內建站台

左邊是自己的目錄，右邊則是 FTP server 的目錄，只要點選兩個視窗中間的箭頭就可以上傳或是下載檔案了。

8.5.2 FTP 的另外一個好選擇 – Filezilla

筆者自己常會到 big5 的站台去逛逛，但是一般的 gFTP，是沒辦法上 big5 網站（若是 PCMan patch 過的就可以）。雖然目前的 Filezilla 3 for Linux 還是測試版，但是該有的功能都有啦！

FileZilla 3 beta 網址

<http://filezilla-project.org/nightly.php>

可以用這個網址先將 FileZilla3.tar.bz2 抓下來，解壓縮到 ~/program 下就可以執行 Filezilla3 了。

📄 下載並安裝 FileZilla 3 beta

先在家目錄裡建一個 program 的資料夾，為了方便管理，然後下載到 program 解開，以下為終端機操作畫面：

```
mkdir program
cd program/
wget -c
http://filezilla-project.org/nightlies/2006-08-06/i386-linux/FileZilla3.tar.bz2
tar xvf FileZilla3.tar.bz2
```

解開後可以看到有一個 FileZilla3 目錄，FileZilla3 的執行檔就放在裡面的 bin 目錄，接下來在終端機下就可以直接執行囉！

```
$ ~/program/FileZilla3/bin/filezilla &
```

啟動後會有個提示跟你說，這是 beta 版本所以可能會有 bug 與一些風險存在，請不要跟他們抱怨，這裡不要理它，按**確定**就可以了。

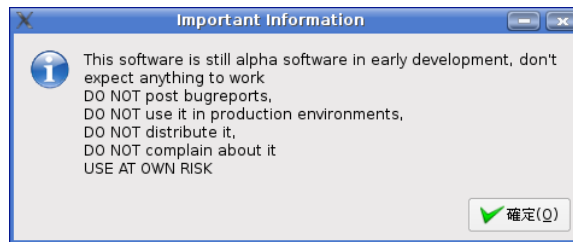


圖 8.101 Important Information

程式就跟著開啟了，要是你之前有設定讀取的編碼是 big5，那麼會有個 Error 出現不能將檔案名稱轉換成 Unicode，不過也沒關係按下確定就可以了。

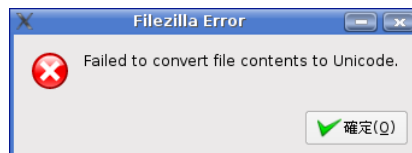


圖 8.102 Filezilla Error

FileZilla 就順利執行囉！

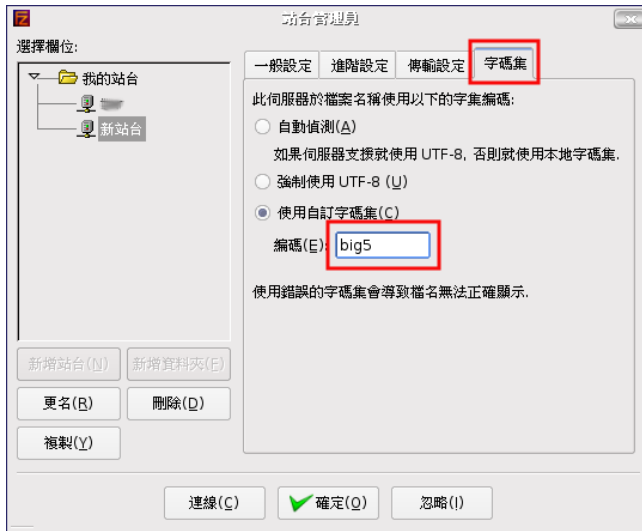


圖 8.103 FileZilla

那麼重點來了，怎麼新增站台並讓 FileZilla 可以自動轉編碼成 big5 呢？開啟站台管理員，檔案--> 站台管理員，然後再這裡點新增站台，接下來輸入 server 的 IP 位址，如果需要帳號密碼的話，登入型式請選一般。

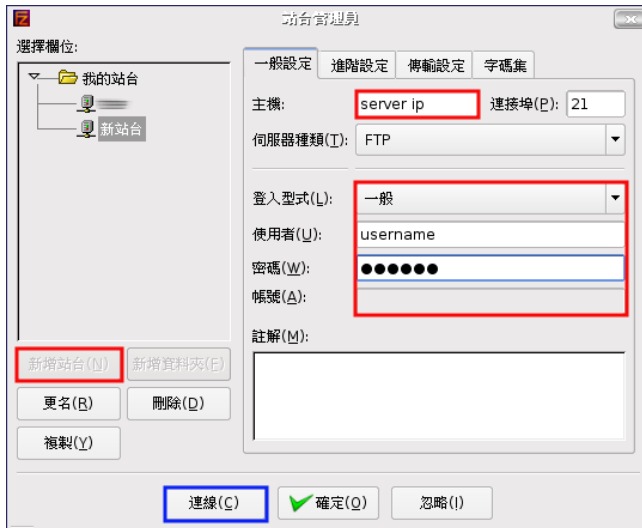


圖 8.104 站台管理員

然後再切換到字碼集，使用自定字碼集，接著輸入 Big5 就大功告成了。

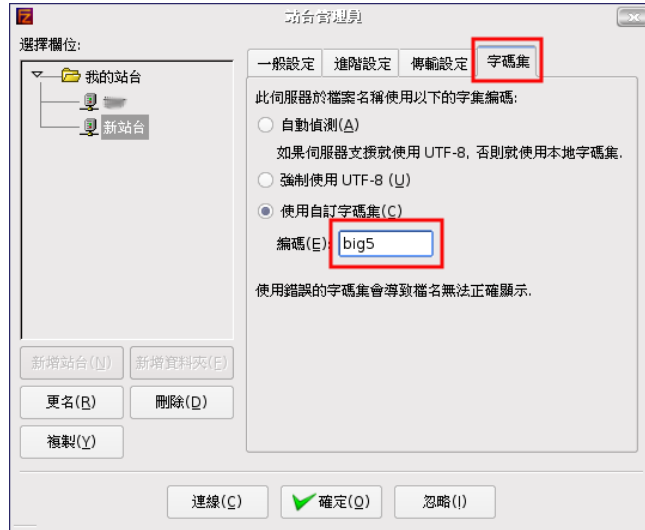


圖 8.105 設定讀取 big5

然後按連線就可以下載東西啦！

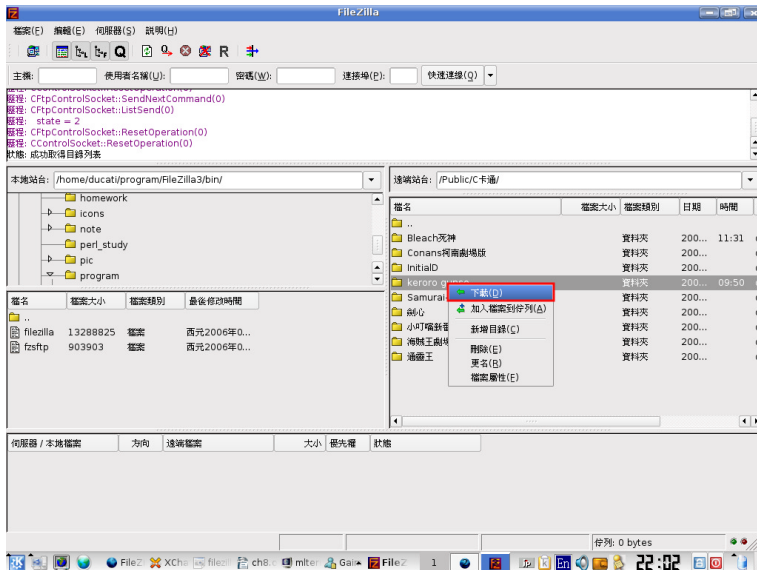


圖 8.106 下載

8.5.3 Linux 上的 FlashGet – Downloader for X

Downloader for X 的介面很像 Windows 上的 FlashGet，使用方式也差不多，而在 ubuntu 上安裝也很簡單，也是用 apt-get 就可以了。

```
$ sudo apt-get install d4x
```

裝好後在**應用程式--> 網際網路--> Downloader for X** 啟動或者是在 console 下輸入 `d4x &` 也可以執行。

```
$ d4x
```

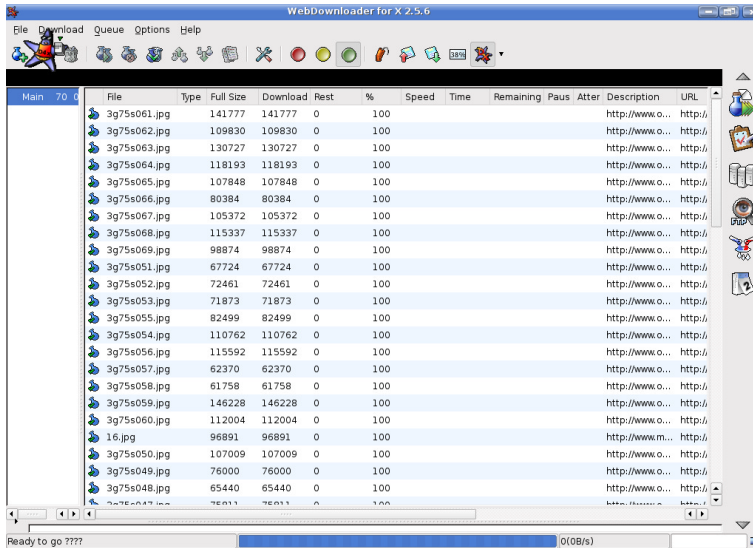


圖 8.107 Downloader for X

接下來我們可以找一個都是圖的網頁來試試，我們只要將想要下載的東西或圖拖曳到左上角的 d4x 星星人小圖即可



圖 8.108 拖曳到下載區

接著會出現下載對話框，不過d4x會自動幫你下載到家目錄下的MyDownloads

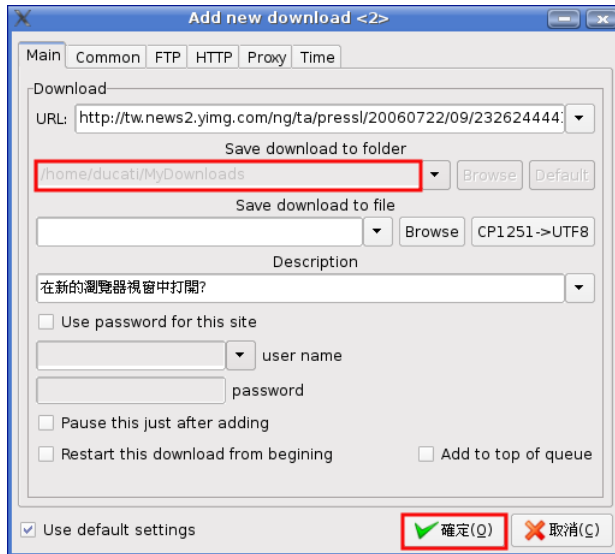


圖 8.109 下載對話框

修改下載目的地目錄，開啟 Queue--> Properties，接著出現 Options 對話框，我們選擇 Download，然後輸入你想要下載檔案回來的目的地。

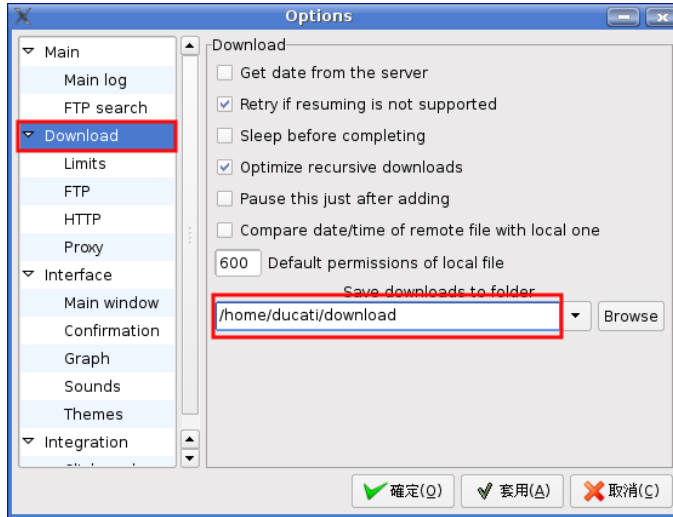


圖 8.110 修改下載目錄

不過其他還有一些過濾功能等等，也可以自己試試看。

8.6 P2P 檔案分享

P2P 的檔案分享方式是目前網路上最常見的，許多如一些 Bt 下載的程式，或是很多要付費的音樂下載軟體也是用 P2P 的方式傳檔，所以在 Linux 上免不了也一定也會有 P2P 的軟體囉！那麼接下來我們介紹兩個 Linux 上的 P2P 軟體，怎麼安裝設定與使用。

8.6.1 好用的 BT 下載程式－Azureus

安裝 Azureus

```
$ sudo apt-get install azureus
```

然後在 應用程式--> 網際網路--> Azureus，就可以開啟囉！接下來會有一連串的提升：

先選擇語言我們選 zh(tw)，意思是繁體中文。



圖 8.111 選擇語言

接下來的配置選項先用 Beginner



圖 8.112 配置選項

選擇連線速率的地方，用 adsl/cable xxx/256 kbps 就可以了。

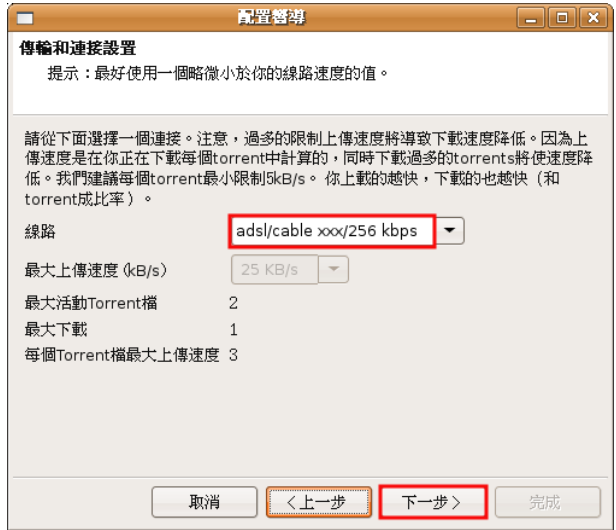


圖 8.113 選擇連線速率

然後會提示你想用哪一個 port 來連線，用預設的就可以，或是你要改成別的也可以，還有一個測試按鈕，可以測試看看你的 NAT port 有沒有通。

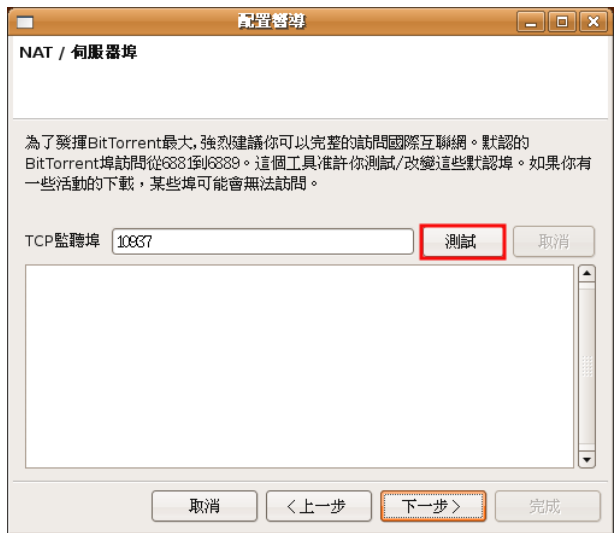


圖 8.114 測試 NAT port

最後是選擇 .torrents 檔案要存放在那裡，筆者建議用預設值即可，方便日後找檔。

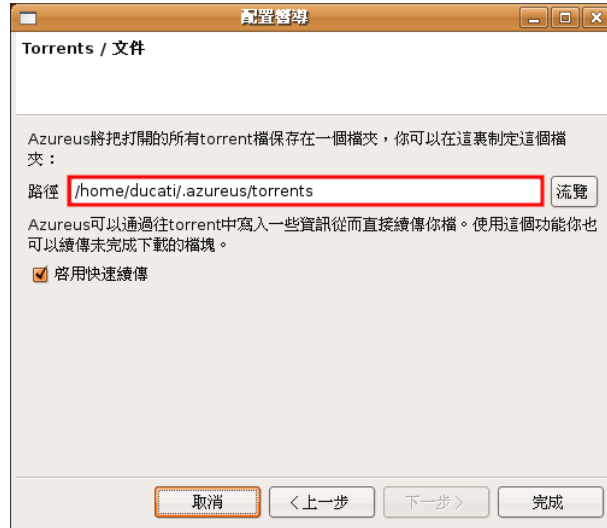


圖 8.115 .torrents 檔案路徑

接下來有個歡迎提示，直接 close 掉。



圖 8.116 歡迎提示

然後我們可以用 ubuntu 官方提供的 ubuntu bt 種子來試試。

網址 <http://ftp.ncnu.edu.tw/Linux/ubuntu/ubuntu-cd/6.06/ubuntu-6.06-alternate-i386.iso.torrent>

下載完後，在主程式文件選項的下方，有一個資料夾將它開啟，選擇剛剛下載的 .torrent 檔案，並選擇存放在哪裡就可以下載了。

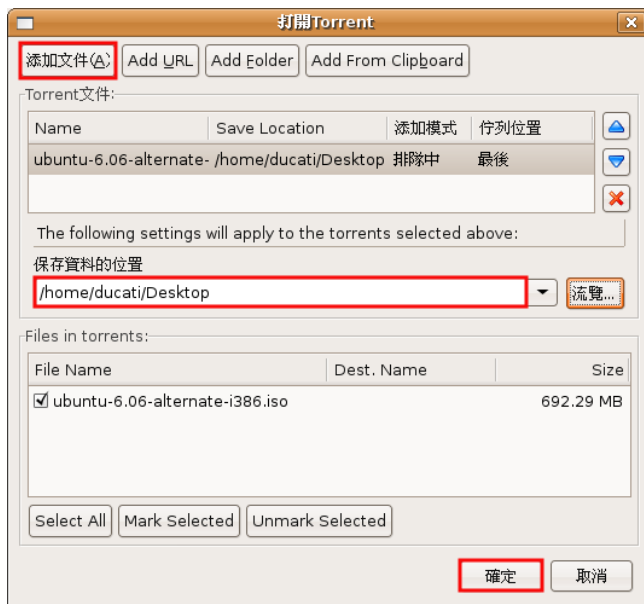


圖 8.117 開啟 .torrent 並下載

BT 檔案下載中囉！

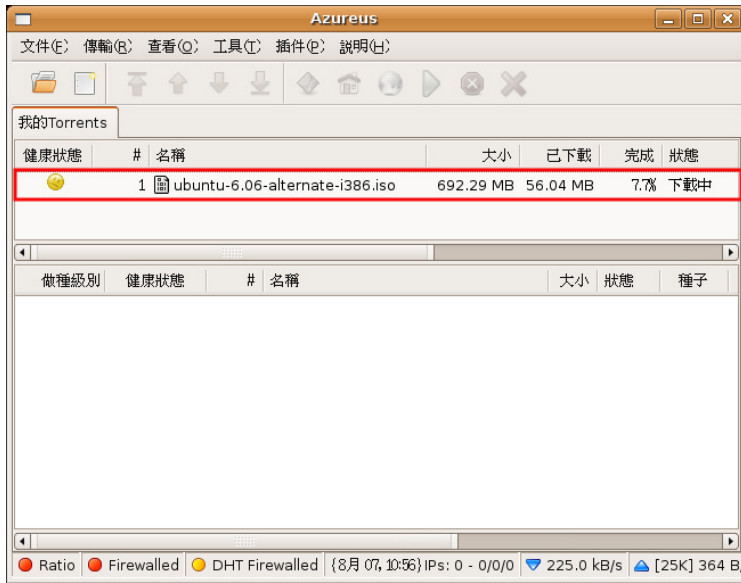


圖 8.118 檔案下載中

8.6.2 Linux 下的騎驢（eMule）程式－aMule

aMule 可以說是真的 eMule，只是 Linux 版的名字不一樣而已，就如同 apt-cache search amule 的解釋一樣，amule - aNOTHER eMule P2P Client。

```
$ sudo apt-get install amule
```

接下來一樣是在 **應用程式**--> **網際網路** 裡找到 aMule，開啟它。接著會看到一個訊息跟你說，這是你第一次使用 aMule，你可以透過哪個 irc channel、以及網址與研發團隊聯繫，以及在哪個論壇回報 bug 等等，按確定就可以了。



圖 8.119 aMule 訊息

開啟後程式的左上角有一個伺服器，旁邊有個藍色的箭頭按下去，aMule 就會自動幫你更新伺服器列表。

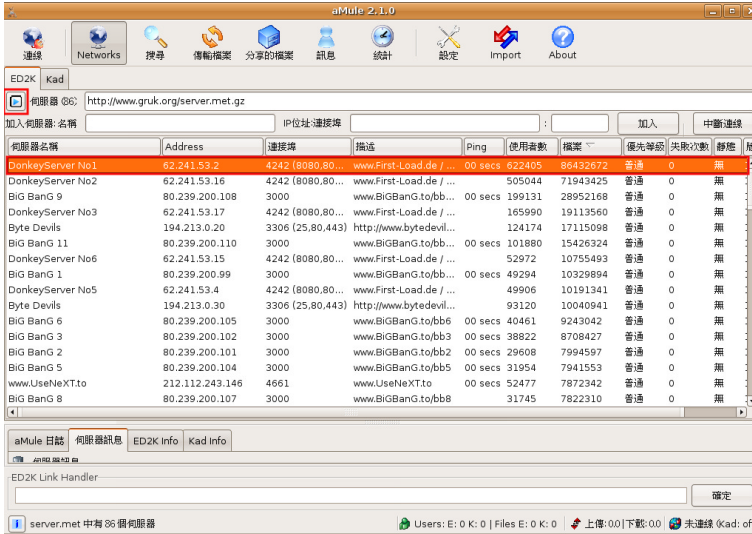


圖 8.120 更新伺服器列表

接著我們在檔案數最多的伺服器上點兩下就可以連線，然後在下面的伺服器訊息可以看到歡迎訊息喔！

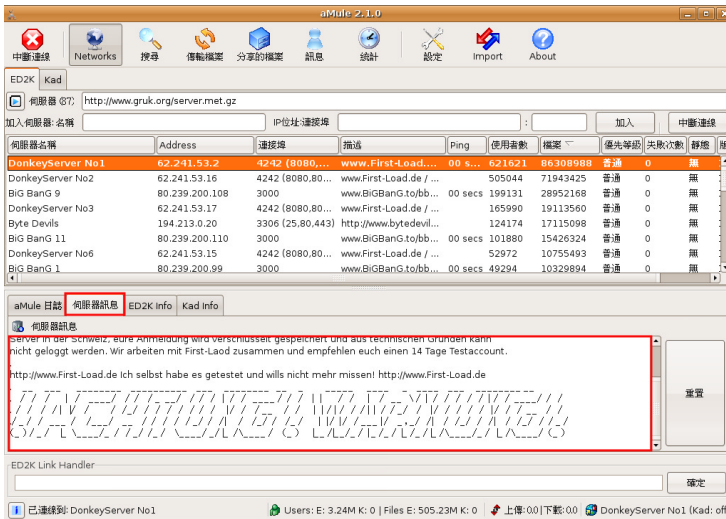


圖 8.121 連線到伺服器

然後測試搜尋一下，看是不是有東西，關鍵字 ubuntu

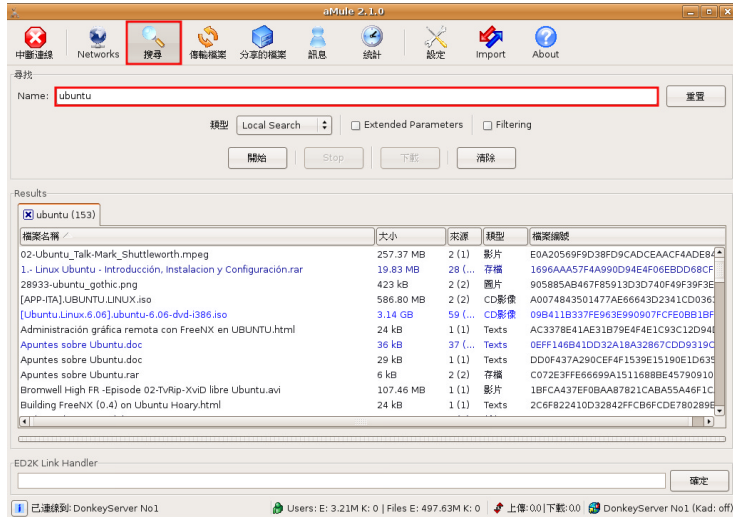


圖 8.122 關鍵字搜尋

看到一大堆跟 ubuntu 有關的，下載 ubuntu 6.06 i386 的 DVD iso 檔試試，按右鍵就可以下載了。

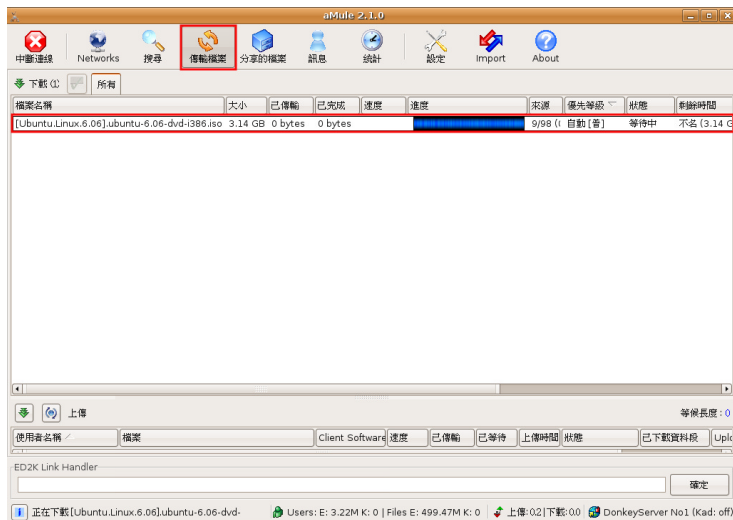


圖 8.123 檔案傳輸

也可以更改下載目錄喔！只要在程式的設定點開視窗就可以改了，記得要在選擇目錄的地方更改。

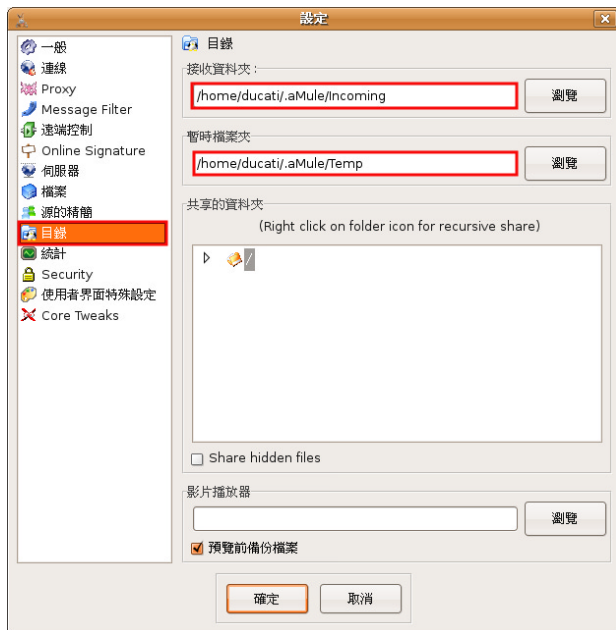


圖 8.124 更改下載目錄

CHAPTER

9

多媒體影音欣賞

說實在地，筆者剛開始使用 linux 底下的多媒體滿不習慣的，有些東西要設定有些則不用。為什麼呢？因為 free 的東西就是讓你摸索找出最適合自己、最好用的，通常上手後不會想再去用其他的軟體，因為真的很方便，甚至可以使用一種多媒體播放器看所有檔案格式的影音了，以下就讓筆者來為大家介紹好用的多媒體播放器，基本上分成聽音樂的、播放影片這兩類來跟你介紹。

9.1 一些額外的多媒體支援安裝

嘿嘿！有了這些招術想看什麼影片或是媒體都不是什麼問題囉！而且可以一次解決不能看 DVD 和 wmv、rmvb 等等的困擾。

9.1.1 破解萬惡的 DVD 影片區碼

相信大家都有過一種經驗，每每當你開開心心地去租了影片回來正準備觀賞時，卻要你選擇 DVD 的影片區碼！並且會提示你「請選擇區域碼 剩餘選擇次數 . . .」。天啊！這不是很煩嗎？由於萬惡的八大影業，為了防止盜版影片的流行，所以把 DVD 區碼分成 1 到 6，而台灣屬於區碼 3 的地方。然而在 Linux

底下的播放軟體不能解 DVD CSS（版權因素），所以導致市面上原版的 DVD 影片反而在 Linux 上不能播放，當然這件事早就被造福人群的駭客破解了。

目前有兩種方法，方法一較不自動，但是隨著 ubuntu 改版應該皆能使用，而方法一是用 ubuntu 下的一個自動安裝 script 做的事情和方法二相同。所以讀者先試試看方法一吧！若失敗了再試試看方法二。

方法一：

```
$ sudo sh /usr/share/doc/libdvdread3/install-css.sh
```

方法二：

```
$ sudo gedit /etc/apt/source.list
```

加入

```
deb-src http://download.videolan.org/pub/videolan/debian sarge main
```

這是借用 debian 的 source code，而為什麼不要用上面提供的 binary 呢？因為我是在 ubuntu 上，怕會有 lib 出問題，且我 notebook 是用 ibook g4，根本沒有 binary package。

然後

```
$ sudo apt-get build-dep libdvdcss2
```

來安裝相關編譯所需程式。

現在要取得 source code 了，並且編譯

```
$ sudo apt-get source libdvdcss2
```

抓完 source code 後，它會自動解壓縮，在我的機器上多了一個目錄 libdvdcss-1.2.8

```
$ cd libdvdcss-1.2.8; dpkg-buildpackage
```

它就會開始編譯啦！（目錄名稱可能會不一樣喔，因為版本的不同）。

然後再回到上一層目錄，應該有幾個編好的 package 吧？就裝上吧！

```
$ sudo dpkg -i libdvdcss*.deb
```

然後你就可以看正版片啦！很諷刺吧，反而盜版的不用那麼麻煩...XD

9.1.2 安裝 Win32 Codecs 影片解碼器

相信大家在 Linux 上都有不能看 wmv 影片檔的經驗，下面我們將介紹怎麼安裝 Win32 Codecs 影片解碼器。安裝這個影片解碼器也可以讓我們在看 rm 或是 rmvb 檔的時候較順（其實本來是完全不能看的）！而且裝了之後幾乎任何檔案格式的影片都可以看哦！聽起來是不是很迷人呢？接下來我們就介紹一下怎麼安裝。不過在這裡要注意一下，這個 Codecs 在某些國家是不合法的，使用前請讀者先注意是否合法，可能會有授權的爭議以及侵權疑慮。

在這裡我們就直接用 wget 抓 ubuntu 台灣站的 .deb 檔下來安裝。

```
$ wget -c http://apt.ubuntu.org.tw/ubtw-testing/w32codecs/w32codecs_20050412-0.4_i386.deb
```

當你抓下來後就可以在目前的目錄看到，假如說你的檔案抓不下來，有可能是新的 codecs .deb 檔出來了所以抓不到，不過讀者也可以先上 ubuntu apt mirror 站看看有沒有出新檔再抓。網址 <http://apt.ubuntu.org.tw/ubtw-testing/w32codecs/> 裡面有個大約 13MB 的 .deb 檔就是囉！

接下來將 win32codecs 的 .deb 檔裝起來，由於我們抓下來的檔案是 .deb 檔所以我們用 dpkg 來安裝！

```
$ sudo dpkg -i w32codecs_20050412-0.4_i386.deb
```

接下來即可試試是否能看 wmv 的檔案，即使是原先完全不能看的 .rm 或是 rmvb 影片檔，用筆者的 x20 用來看也順多了。

9.2 Mplayer – Linux 中支援最多格式的軟體播放器

Mplayer 在 Linux 上是很有名的影片播放軟體，不但輕巧好用，而且具有簡單直覺化的快速鍵，只要輕按一個按鍵就可以任意控制 Mplayer 的播放行為，例如：PageUp 按一下就可以選擇上一個段落、PageDown 則是往後一個段落，上是四倍快轉、右是兩倍快轉，下和左則反之，f 則是 fullscreen 全螢幕等等，既方便又好用！

9.2.1 使用 Mplayer 播放有字幕檔的 avi 影片

如果你曾想在 Linux 上播放影片而去找過資料的經驗就知道它多有名囉！功能很強大，可以播放的影音格式超多，包括 VCD DVD MPG RMVB 等等。播放的方式可以用圖型 UI，也可以在 terminal 下播放哦！不過筆者都是在 terminal 播放為多，這是習慣問題啦！在 ubuntu 下安裝完可以直接在 **應用程式**--> **影音**--> **Mplayer Movie Player** 可以直接執行。

安裝主程式

```
$ sudo apt-get install mplayer-386
```

原則是這樣安裝就可以有主程式了，不過讀者也可以用 apt-cache search 搜尋一下另外的套件安裝

```
$ sudo apt-cache search mplayer
```

```
mplayer - The Ultimate Movie Player For Linux
mplayer-386 - The Ultimate Movie Player For Linux (dummy package)
mplayer-586 - The Ultimate Movie Player For Linux (dummy package)
mplayer-686 - The Ultimate Movie Player For Linux (dummy package)
mplayer-amd64 - The Ultimate Movie Player For Linux (dummy package)
mplayer-custom - The Ultimate Movie Player For Linux (dummy package)
mplayer-doc - The Ultimate Movie Player For Linux (Documentation)
mplayer-fonts - Fonts for mplayer
mplayer-g4 - The Ultimate Movie Player For Linux (dummy package)
```

```
mplayer-k6 - The Ultimate Movie Player For Linux (dummy package)
mplayer-k7 - The Ultimate Movie Player For Linux (dummy package)
mplayer-nogui - The Ultimate Movie Player For Linux
mplayer-powerpc - The Ultimate Movie Player For Linux (dummy package)
mplayer-skins - Skins for the Ubuntu mplayer Package
mozilla-mplayer - MPlayer-Plugin for Mozilla
```

其實還有很多版本，讀者也可以選擇適合自己機器的版本來安裝。

接下來安裝一下 mplayer 的字型以及 mozilla 的外掛吧！

```
$ sudo apt-get install mplayer-fonts          字型
$ sudo apt-get install mozilla-mplayer      mozilla 的 mplayer plugin
```

接下來更改設定檔，先備份一下。

```
$ sudo cp /etc/mplayer/mplayer.conf /etc/mplayer/bak.mplayer.conf
```

更改設定檔

```
$ sudo gedit /etc/mplayer/mplayer.conf
```

找一下這一行

```
vo=x11,          # To specify default video driver
```

然後改成

```
vo=xv,          # To specify default video driver
```

這樣基本安裝就 ok 囉！而且可以在 mozilla 觀看網頁，有嵌入影片時也可以播放 mplayer 支援的影片格式。

接下來進入正題，使用 Mplayer 播放有字幕檔的 avi 影片。字幕檔格式常見的有兩種，第一種是 .srt、第二種是 .sub，我們先來介紹一下如何掛上 srt 的字幕，假如今天我們的電影檔名稱是 kill-b.avi、字幕檔為 kill-b.srt，開啟 mplayer 應用程式--> 影音--> Mplayer Movie Player。



圖 9.1 Mplayer GUI MODE

接下來我們在 mplayer 的功能選單按滑鼠右鍵選 **preferences**，按照以下設定將 Mplayer 設定讀取 big5 編碼的字幕檔、及 truetype 字型，就可以將 big5 編碼的 .srt 字幕掛上去。

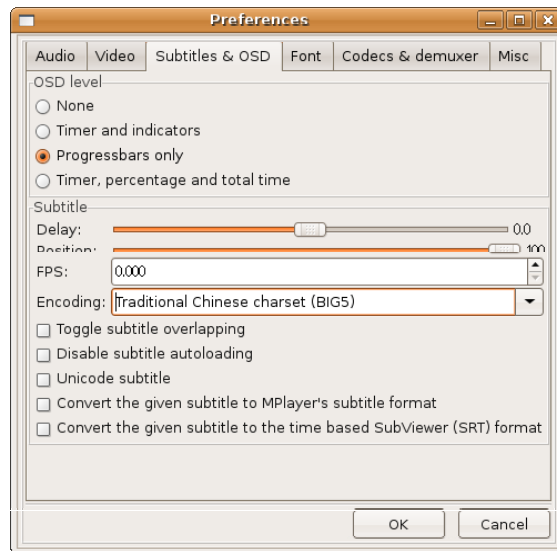


圖 9.2 設定 Mplayer 讀取 big5 編碼的字幕檔

設定好後，只要是字幕檔的檔名和影片檔的檔名是一樣的，即可以播放有字幕檔的 avi 影片囉！

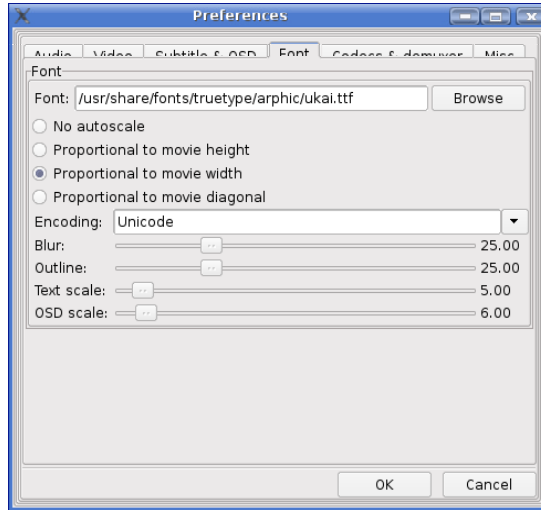


圖 9.3 設定 Mplayer 字型檔

字型檔也設定一下，讓 Mplayer 選擇你想要用的字型來播放檔案。

9.2.2 使用 Mplayer 播放 DVD VCD

開啟 Mplayer GUI Mode 應用程式--> 影音--> Mplayer Movie Player，然後在 mplayer 的功能選單按滑鼠右鍵選，就可以選擇播放 DVD 或是 VCD 哦，而且也可以選擇播放的語言、字幕語言、段落以及全螢幕等等，應有盡有。

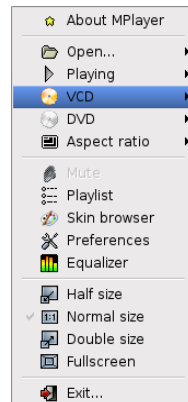


圖 9.4 mplayer 細部功能選擇

若要用 Mplayer 看 DVD 或是 VCD 的話，在功能視窗上按右鍵即可選擇，除了有 DVD、VCD 等可以選擇外，你還可以設定影片檔案的來源、播放列表、播放網址、中英文字幕、或是是否全螢幕等都可以設定。當然囉，功能不只有這樣！

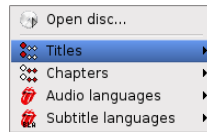



圖 9.5 DVD 字幕、語言、段落選擇

也可以選擇字幕、語言、段落等等，應該有的功能一樣都不會少。

 讀者也可以在 console 下播放影片

```
$ mplayer 影片名稱
```



圖 9.6 在 console 下播放影片

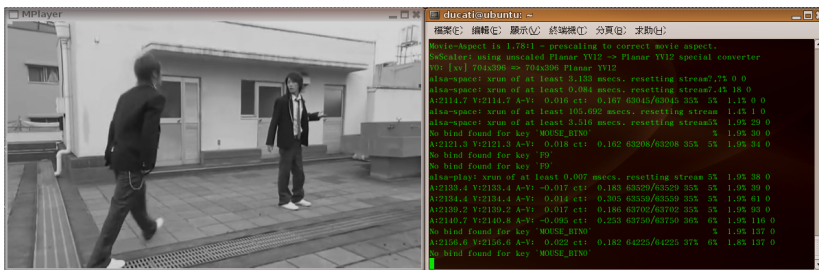


圖 9.7 看片的同時也可以在 console 看到影片的資訊

對了，我提醒一下，若播放 DVD 有水波紋的話，可以試著去交錯打開（Deinterlace）喔！這是因為有些 DVD 錄製的時候是用交錯視訊號，所以才會有水波紋呢！

9.3 XINE — 介面類似 PowerDVD 的多媒體播放器

它有好用的操作介面！還有像 PowerDVD 的介面讓人覺得使用上也蠻平易近人的，使用 XINE 播放 DVD 的話只要將光碟片放入後按一個按鈕就可以了哦！

✎ 安裝 xine

安裝的時候，我們要連帶安裝 xine 的 codecs（解碼器）才可以看 XviD codec 的影片，因為 avi 檔會用到，接下來用 command 來將 xine 的 user interface 和 xine codecs 裝上去

```
sudo apt-get install xine-ui libxine-extracodecs
```

安裝到這裡的時候就可以看 DVD 和 VCD 了，很簡單吧！ubuntu 裝軟體就是這麼方便，若要看有字幕檔的 avi 影片請讀者接下去看。



圖 9.8 xine user interface 使用者介面

xine 的控制界面

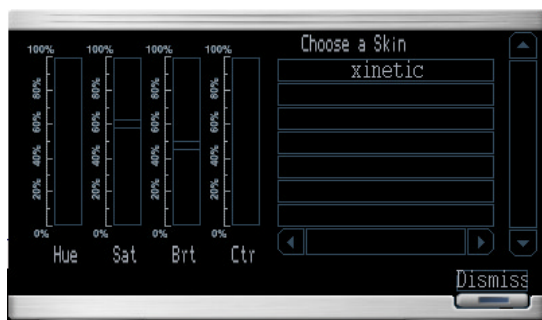


圖 9.9 xine controller

xine 的 play list

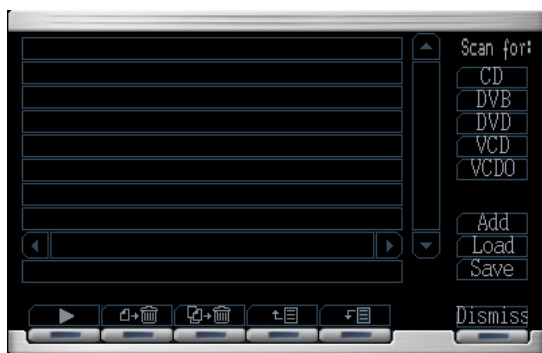


圖 9.10 xine playlist

9.3.1 使用 XINE 播放有字幕檔的 avi 影片

xine 可以掛載 .srt 格式的字幕檔，但是在目前正常情況下安裝 xine 是不會自動幫你掛字幕檔上去的。因為 xine 預設安裝的字型是沒有中文字型，不過我們都幫你考慮進去囉！所以可以不必像網路上一樣去抓 xine-lib 下來編譯 xine-fontconv 來轉字型檔，只要把筆者幫你轉好的 xine 的中文字型，放到 /usr/share/xine/libxine1/fonts，然後再設定家目錄底下的 .xine/config，就可以掛載字幕看 .avi 的影片囉。那麼先請讀者將筆者轉好的字型檔先抓下來吧！

```
$ wget -c http://www.cqis.ncku.edu.tw/~dbtsai/xine-akai.tar.bz2
```


抓下來後可以開啟目錄視窗直接在檔案按右鍵點選解壓縮檔案，或者是直接下 command 來解開檔案

```
$ tar jxvf xine-akai.tar.bz2
```

解開壓縮檔後將裡面的檔案放到 `/usr/share/xine/libxine1/fonts`

```
$ sudo cp akai-*.xinefont.gz /usr/share/xine/libxine1/fonts
```

接下來修改 `~/xine/concig`

```
$ gedit ~/.xine/config
```

在裡面加入

```
# tiny small normal large very large huge
# 這是用來對應到剛才下載解壓縮後的六個大小的字型，
subtitles.separate.subtitle_size:large
# 用全螢幕看電影差不多用 large 就可以了，如果覺得不夠大也可以用 huge 試看看
subtitles.separate.vertical_offset:40
# 這裡是設定字體離視窗最下面的邊緣距離，如果讀者是用全螢幕看的話差不多是 40
subtitles.separate.font:akai
# 這是剛剛自訂的字型名稱
subtitles.separate.src_encoding:big-5
# 字型編碼用 big-5 通常在網路上下載回來的都是 big-5
```

這樣子就可以看有字幕檔的 .avi 影片囉！

9.3.2 使用 XINE 播放 DVD 和 VCD

還記得我們上面說的嗎？xine 是一個滿像 windows 下 powerDVD 的介面哦！不過當然是不一樣的東西，老實說剛開始使用有點不習慣，不過使用過幾次後就很上手了，當然 xine 可以很簡單的播放 DVD 或是 VCD 光碟，因為又加上了 ubuntu 自動地幫你掛載上你的多媒體硬體（例：DVD 光碟機、DVD 燒錄機、一般的光碟機、燒錄機等等的）。所以讀者你只要將光碟片放入你的電腦後開啟 XINE 按下它的 DVD icon、VCD icon 就可以播放了，請點選下圖用紅色框框選起來的地方就可以囉！



圖 9.11 使用 XINE 播放 DVD、VCD

若播放 DVD 有水波紋的話，要和 Mplayer 一樣打開去交錯喔！

9.3.3 Gxine – Gnome 介面的 xine 播放器

Gxine 跟 xine 相比，較不會那麼複雜。用滑鼠點兩下就可以看 DVD、VCD 或檔案，還有線上廣播哦，下面是 Gxine 的安裝以及簡單的介紹：

✎ 安裝

```
$ sudo apt-get install Gxine
```

✎ 應用程式--> 影音--> Gxine Movie Plyaer

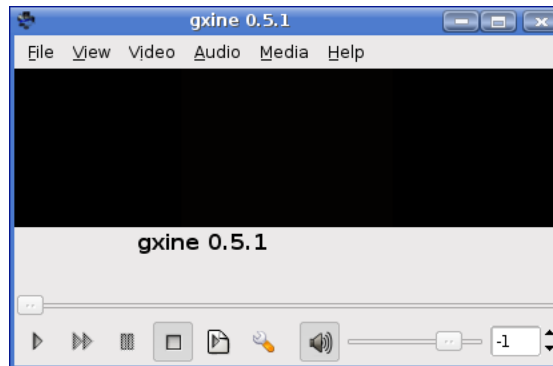


圖 9.12 Gxine

Gxine 播放 DVD、VCD 也很簡單哦！
將 File 這個選單拉下來就可以了。

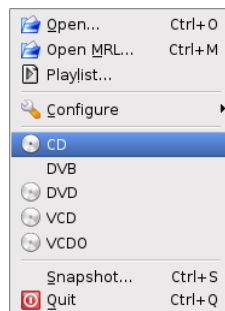


圖 9.13 選擇要播放的檔案或是媒體

還有一些觀看的設定例如：視窗的放大縮小、字幕的選擇、全螢幕的模式等等。

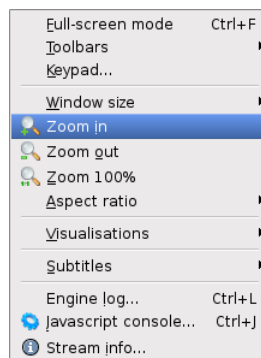


圖 9.14 Gxine 觀看設定

也可以收聽線上廣播哦，喜歡的頻道可以加入 media mark。

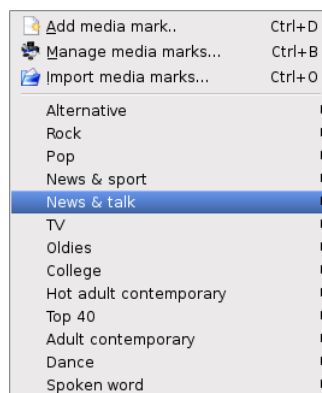


圖 9.15 Gxine 聽線上廣播

9.4 VLC — 跨平台的多媒體播放器

vlc 是一個很多功能的多媒體播放器，裝好後，你想不到的功能也都會有，解碼器也很完整，但設定程序很繁雜就是了；好在不用另外再開 console，直接在設定--> Preference 就可以設定了，接下來我們要設定四個地方，依照下面的圖示設定就可以囉！



圖 9.16 vlc 設定

9.4.1 自動偵測字幕檔



圖 9.17 自動偵測字幕檔

如果 vlc 沒有幫你找到字幕檔，或是你的字幕檔沒有跟影片檔放在一起的話，可以用指定的方式，在**使用字幕檔**的地方輸入你的檔案所在地即可。

9.4.2 設定自型檔和字體大小

在這裡我們使用內建的字型即可，或是你想選擇自己喜歡的，重要的是可以播中文，字型檔路徑 `/usr/share/fonts/truetype/arpic/ukai.ttf`，接下來設字體大小，因為筆者用 x20 12.1 吋來看時，預設的字型有點小、會模糊看不清楚，所以在這裡筆者用 60 大小是差不多的，你也可以依自己喜好來調整。



圖 9.18 設定自型檔和字體大小

9.4.3 選擇字幕格式



圖 9.19 選擇字幕格式

讓 vlc 為你判別字幕檔是什麼格式。

9.4.4 選擇字幕編碼



圖 9.20 選擇字幕編碼

因為通常下載來的字幕檔都是 big5，除非是有轉檔過，所以這裡我們選擇 big5 編碼。別忘了修改完後，要記得儲存或是重新開啟 vlc 哦！




圖 9.21 vlc 播 avi 影片

9.5 Totem — 預設的多媒體播放器

Totem 是在 ubuntu 下預設就裝好的多媒體播放器，如果你安裝的是 ubuntu 而不是 kubuntu、xubuntu 的話，預設的 ubuntu CD 裝起來就有了，不過呢！在你使用之前必需要先為 Totem 選擇一個播放引擎，以下進入正題囉！

9.5.1 使用 Gstreamer 當播放引擎

GStreamer 是一個串流的媒體架構，用來處理多媒體資料。透過使用這個函式庫應用程式可以做聲音處理、演奏錄影。它的架構是使用外掛的方式載入，所以新資料類型或程序可由簡單安裝來增加外掛播放。所以在下面我們要安裝一些 plugin 來處理我們想看的多媒體資料。

 將 Totem 的播放引擎換成 gstreamer

```
$ sudo apt-get install totem-gstreamer
```

然後用 apt search 一下我們要裝的 plugin

```

ducati@X20:~$ apt-cache search gstreamer0.10-plugins | more
gstreamer0.10-plugins-base - GStreamer plugins from the "base" set
gstreamer0.10-plugins-base-apps - GStreamer helper programs from the "base" set
gstreamer0.10-plugins-base-dbg - GStreamer plugins from the "base" set
gstreamer0.10-plugins-good - GStreamer plugins from the "good" set
gstreamer0.10-plugins-good-dbg - GStreamer plugins from the "good" set
gstreamer0.10-plugins-good-doc - GStreamer documentation for plugins from the
"good" set
gstreamer0.10-plugins-bad - GStreamer plugins from the "bad" set
gstreamer0.10-plugins-bad-doc - GStreamer documentation for plugins from the
"bad" set
gstreamer0.10-plugins-base-doc - GStreamer documentation for plugins from the
"base" set
gstreamer0.10-plugins-ugly - GStreamer plugins from the "ugly" set
gstreamer0.10-plugins-ugly-dbg - GStreamer plugins from the "ugly" set
gstreamer0.10-plugins-ugly-doc - GStreamer documentation for plugins from the
"ugly" set
gstreamer0.10-plugins-ugly-multiverse - GStreamer plugins from the "ugly" set
(Multiverse Variant)
gstreamer0.10-plugins-bad-multiverse - GStreamer plugins from the "bad" set
(Multiverse Variant)

```

然後把必要的 plugin 都先裝上去 (good、bad、ugly)，為什麼要把 plugin 裝上去呢？請看以下說明：

- **good**：裝 good 是因為沒有版權問題。幾乎都是 GPL 授權的解碼器或者沒有版權疑慮，good 可以播放 flac、wav、CD、DV、Smoke、jpeg image 等格式。
- **bad**：雖然可以播放，但是因為用逆向工程，所以它們沒有辦法拿到規格，可能因此有 bug 向他們抱怨，與其抱怨，不如提供 patch。可以播放的媒體檔案格式有 .dat (VCD)、Free AAC、MOD、GSM encoder/decoder 等等。
- **ugly**：該媒體編碼方式是有專利限制，需要經過授權才能使用。用這種逆向工程的手法，並不一定是在每個國家都是合法的，可以播放的格式有 MP3 等格式。

```
$ sudo apt-get install gstreamer0.10-plugins-good gstreamer0.10-plugins-bad  
gstreamer0.10-plugins-ugly
```

至少一定要把 good、bad、ugly 裝上去，接下來呢，我們再將其他的 codecs plugin（解碼外掛）裝上去

```
$ sudo apt-get install gstreamer0.10-ffmpeg gstreamer0.10-fluendo-mp3  
gstreamer0.10-fluendo-mp3demux
```

加上上面安裝的總共有六個 plugin，這樣就幾乎所有的影片格式都可以看了。

還記得我們前面裝的 win32codec 嗎？要用 totem-gstream 來看 wmv 這類 windows 的東西記得要先裝 win32codec 後，再透過 gstreamer0.10-pitfdll 來載入 dll 才可以看哦，最後我們還要再裝上 gstreamer0.10-pitfdll

```
$ sudo apt-get install gstreamer0.10-pitfdll
```

接下來我們來試看看播放 .avi 的影片看看



圖 9.22 totem-gstreamer 播 avi

totem-gstreamer 播 wmv



圖 9.23 totem-gstreamer 播 wmv

9.5.2 使用 XINE 當播放引擎

使用 xine 當 totem 的播放引擎就不用設定什麼了，設定檔只要用上面筆者寫的 xine 的設定即可，只不過 Totem-xine 的設定檔放在跟 XINE 不一樣的地方，接下來我們把 totem 的播放引擎換成 xine

```
$ sudo apt-get install totem-xine
```

設定檔是這一個 `~/.gnome2/totem_config` 加入跟上次一樣の設定值

```
$ gedit ~/.gnome2/totem_config
```

加入以下

```
# tiny small normal large very large huge
# 這是用來對應到剛才下載解壓縮後的六個大小的字型，
subtitles.separate.subtitle_size:large
# 用全螢幕看電影差不多用 large 就可以了，如果覺得不夠大也可以用 huge 試看看
subtitles.separate.vertical_offset:40
# 這裡是設定字體離視窗最下面的邊線距離，如果讀者是用全螢幕看的話差不多是 40
```

```
subtitles.separate.font:akai
# 這是剛剛自訂的字型名稱

subtitles.separate.src_encoding:big-5
# 字型編碼用 big-5 通常在網路上下載回來的都是 big-5
```

9.6 RealPlayer

好了～不囉嗦！我們馬上就來把 realplayer 給裝上去吧。

備註 其實 Ubuntu 在 multiverse 有包，不過版本比較舊，若可以忍受的話，直接用 `sudo apt-get install realplayer` 就可以裝了。

1.請先到 realplayer 的官方網站下載

網址 <http://www.real.com/> 你會發現 右邊就有一個 for linux 的下載點,接下來選 free download

2.安裝 Realplayer

因為我們下載回來的是一個執行檔，所以更改權限後可以直接執行安裝

```
$ chmod +x RealPlayer10GOLD.bin
```

更改權限變加上可執行

```
$ ./RealPlayer10GOLD.bin
```

執行在本目錄裡的 RealPlayer10GOLD.bin 這個安裝執行檔，下面是在 console 的執行結果。

```
Extracting files for RealPlayer installation.....

Welcome to the RealPlayer (10.0.7.785) Setup for UNIX
Setup will help you get RealPlayer running on your computer.
Press [Enter] to continue...
```

我們可以在終端機視窗看到歡迎訊息 按下 Enter

```
Enter the complete path to the directory where you want
RealPlayer to be installed. You must specify the full
pathname of the directory and have write privileges to
the chosen directory.
Directory: [/home/ducati/download/RealPlayer]:/home/ducati/program/RealPlayer
```

這時候會要你輸入你要安裝的目錄,筆者習慣上都會在家目錄下再建一個另外安裝軟體目錄 program 或 bin 之類的,方便日後讀者自己管理所安裝的 program。

假如我們想要安裝在這個目錄 /home/ducati/program

將會顯示

```
You have selected the following RealPlayer configuration:

Destination          /home/ducati/program/RealPlayer

Enter [F]inish to begin copying files, or [P]revious to go back to the previous
prompts: [F]: F
```

將大寫 F 給它按下去! 出現以下訊息就表示安裝完成囉!

```
Copying RealPlayer files ...configuring mozilla...
Configuring realplay script...
RealPlayer installation is complete.
Cleaning up installation files...
Done.
```

接下來我們會看到 program 目錄下多了一個 RealPlayer 的目錄

```
$ ls
```

看到 RealPlayer 目錄

```
$ cd RealPlayer}
```

進去看看有沒有執行檔可用!

```
$ ls
```

```
ducati@X20:~/program/RealPlayer$ ls
Bin      common  install.log  LICENSE  plugins  README    realplay.bak  share
codecs  doc     lib          mozilla  postinst  realplay  realplay.bin
```

有耶！我們發現 realplay 是綠色的 哈哈！執行看看吧。

```
$ ./realplay &
```

加上 & 丟到 background 執行。

滿簡單好看的。不過就只有一些基本播放 rm 檔案的功能，但是筆者還是習慣用 Mplayer 看，老實說在爛爛的 x20 上看比較不會 lag，也有 Favorites 等的功能。

如圖 9.24 linux 下的 RealPlayer



圖 9.24 RealPlayer

設定 realplayer 預設的瀏覽器

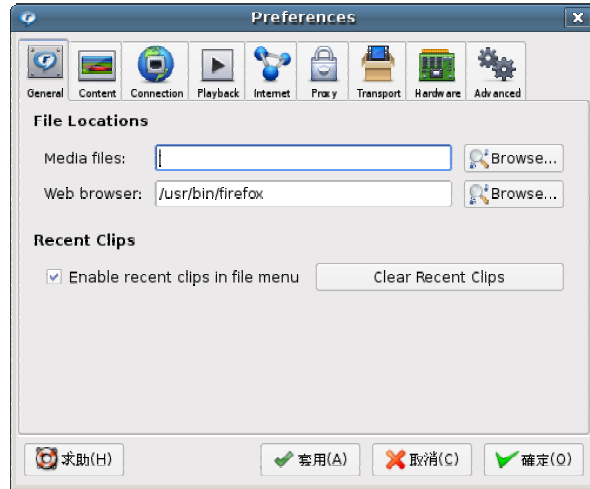


圖 9.25 設定 FireFox 為 RealPlayer 的預設瀏覽器

在 web browser : 那一欄輸入 /usr/bin/firefox 就可以了。

9.7 beep-media-player 源自於 xmms 的 mp3 播放器

beep-media-player 是一個很好用的 mp3 player，之前要是有用過 xmms 的人一定也會覺得不錯用！筆者也都是用這個，介面跟 Windows 下的 winamp 有點類似。

✎ 安裝步驟

```
$ sudo apt-get update;sudo apt-get install beep-media-player
```

執行，在 colsole 下輸入或是 **應用程式--> 影音--> beep-media-player** 都可以執行。

```
$ beep-media-player &
```

『可以直接打 beep 然後按 tab 鍵 shell 會幫你自動把 command 補齊』



圖 9.26 beep-media-player

開啟 beep-media-player4 後要是沒有聲音的話，可以在主程式（不是 playlist）上按右鍵，然後選擇 Preferences-->Plugins，在 outpt 選 eSound Output plugin。



圖 9.27 改為 esound 處理模式

然後我們也可以更改設定讓他支援 big5 中文哦！接下來也是一樣在主程式上按右鍵選 Preferences--> Plugins--> Plugins--> Media，然後在 Media plugin list: 裡選擇 MPEG Audio Plugin，在其偏好設定裡的 Title，將 Convert non-UTF8 ID3 tags to UTF8 打勾，接下來呢只要把我們想要的編碼輸入就可以了！請在 ID3 encoding：這欄輸入 big5。

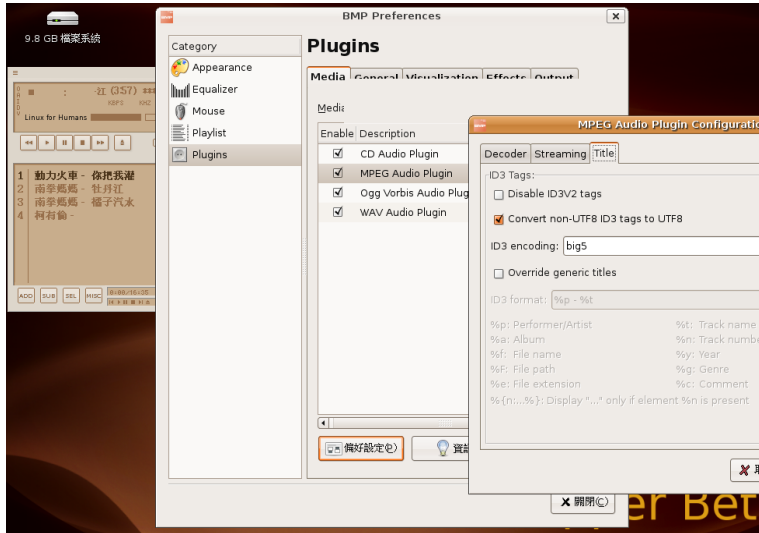


圖 9.28 讓 beep-media-player 可以轉 big5

這樣就可以讀取 big5 的中文檔名了。

9.8 Rhythmbox 收錄 CD 控制 ipod

Rhythmbox 也是 ubuntu 灌好就預設有的，這個軟體很方便，可以將你喜歡聽的 Music CD 收錄到你的電腦裡，也可以用來控制 ipod 哦。由於 Rhythmbox 是用 gstreamer 當解碼器，所以要聽 mp3 的話，要把 gstreamer 外掛裝起來喔。這些外掛名稱在 totem 的 gstreamer 引擎那裡有談到。記得喔！預設 Rhythmbox 是不能放 mp3 的，記得裝外掛。

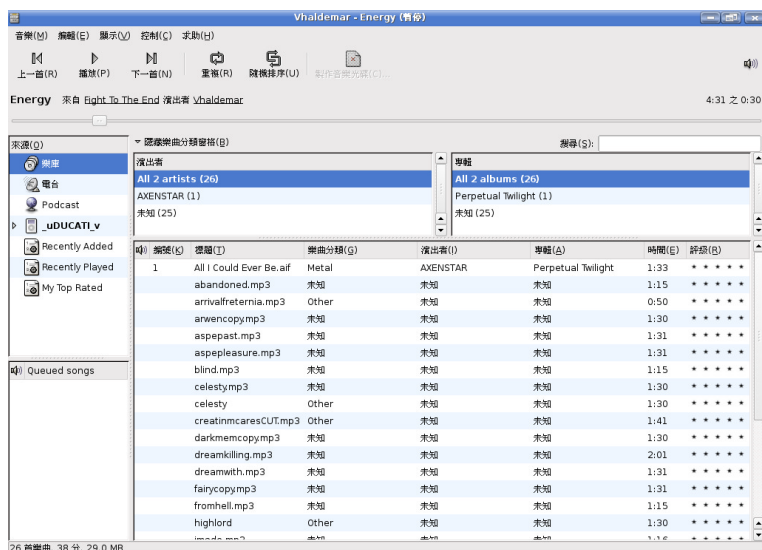


圖 9.29 Rhythmbox

在 Linux 下有個好處就是 ipod 在 Mac 及 Windows 上，裡面的音樂檔案都會保護而隱藏起來，但是在 Linux 下是完全看得到的，只要將你的 ipod 接上 ubuntu 即會自動幫你將 ipod 掛載起來，開啟桌面上的 ipod 資料夾後，音樂是放在 iPod_Control/Music 裡面，然後依照資料裡的編號從 F00 一直到 Fxx 裡面就是 ipod 裡的音樂了，如果讀者照著上面的說明，將 gstream 的那幾個 plugin 裝好後，就可以順利的用 Rhythmbox 來聽 ipod 裡的音樂、也可以將裡頭的檔案一個一個 copy 到 pc 上，首先我們介紹怎麼用 Rhythmbox 來控制 ipod。

使用 Rhythmbox 來控制 ipod

先將 ipod 接上 ubuntu 後，會問你是否將這個裝置掛載起來開啟資料夾或是不，這裡要選擇掛載起來就可以看到 ipod 裡的資料夾，然後開啟 Rhythmbox 選擇 **音樂--> 搜尋可移除媒體** 就可以用 Rhythmbox 控制 ipod。



圖 9.30 搜尋可移除媒體

然後在左邊的來源點選你的 ipod 圖示就可以囉！

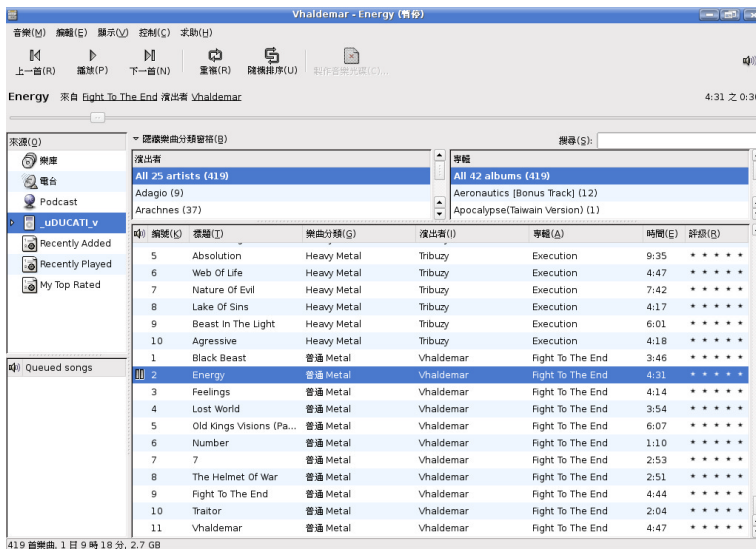


圖 9.31 ipod 裡的音樂列表

不過這只有簡單的讀取功能，如果你要將 ipod 的音樂用軟體匯入或是匯出的話可以使用 gtkpod 來做。

把 Music CD 裡的音樂備份到電腦裡

直接放入 CD，Ubuntu 會為你開啟 Sound juicer，就可以選播放或是讀取囉！



圖 9.32 匯入音樂光碟

讀者也可以裝 gtkpod 來試試。

9.9 amarok

其實，這幾個音樂播放軟體都差不多，所以在此簡單介紹一下 amarok 也可以支援 ipod 的播放，不過它不一樣的是它可以將可移除的媒體（如 ipod）和你電腦裡的檔案整合在一起、也可以燒成 CD，它控制 ipod 的方式是你開啟 amarok 後就會自動載入了，可以在 Media Device 看到。

用 amaroK 讀取 ipod 內容

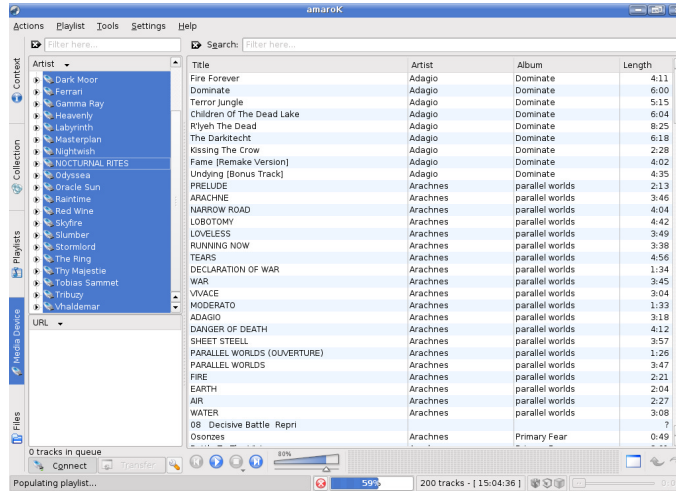


圖 9.33 amaroK 讀取 ipod

將左邊 ipod 的內容選定後，直接拉到右邊的清單就可以播放了

用 amaroK 讀取電腦裡的音樂

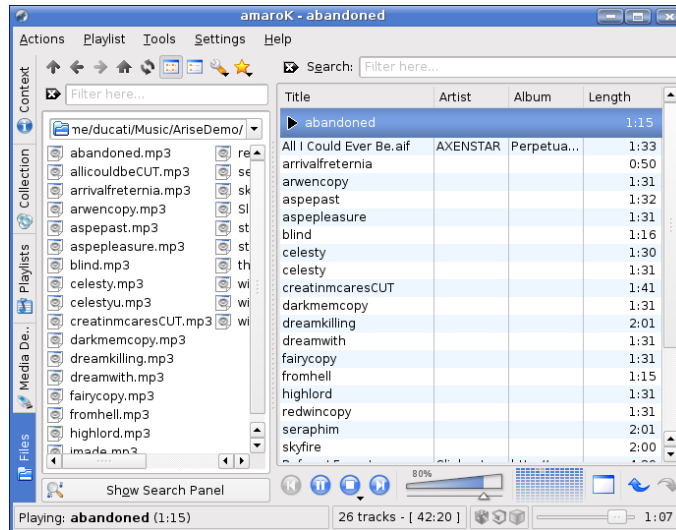


圖 9.34 amaroK 讀取電腦裡的音樂

想要播放 pc 裡的音樂只要在左下方選擇 Files，然後在左上方填入你放音樂的資料夾即可。

用 amarok 將電腦裡的音樂燒出來

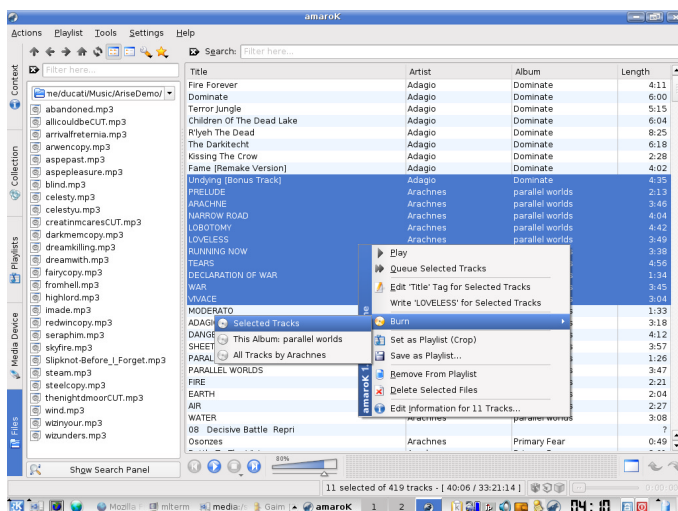


圖 9.35 amarok用 amarok 將音樂燒出來

只要將你的檔案選定後按右鍵選擇 Burn 就可以燒出屬於你的 Music CD

9.10 廣播收聽軟體－Directory Browser (streamtuner)

streamtuner 是很好用的收聽廣播軟體，它有好幾個站蒐集了很多的不同的音樂類型以及不過的播放音樂格式，收集音樂頻道的網站有 Xiph、SHOUTcast、Live365、punkcast.com、Google station、basic.ch、等等，各種類型都有，音樂內容也是五花八門，舉凡古典、爵士、RAP、甚至重金屬都有！Google station 裡有更多的細分，你也可以先在 Google station 裡先搜尋一番後再到 Xiph、SHOUTcast、Live365、punkcast.com 點播。接下來我們來看看如何安裝吧！

```
$ sudo apt-get install streamtuner
```

然後在 Terminal 下執行或是 logout 一下再登入 gnome 也可以在，應用程式--> 影音--> streamtuner 也可以執行。

```
$ streamtuner &
```

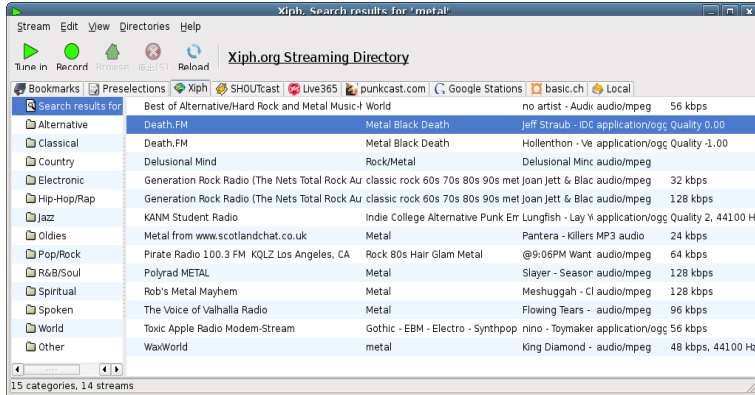


圖 9.36 Streamtuner

接下來我們把預設的播放器改成 beep-media-player 吧！看圖說故事哦！我們先選擇 Edit--> Preferences

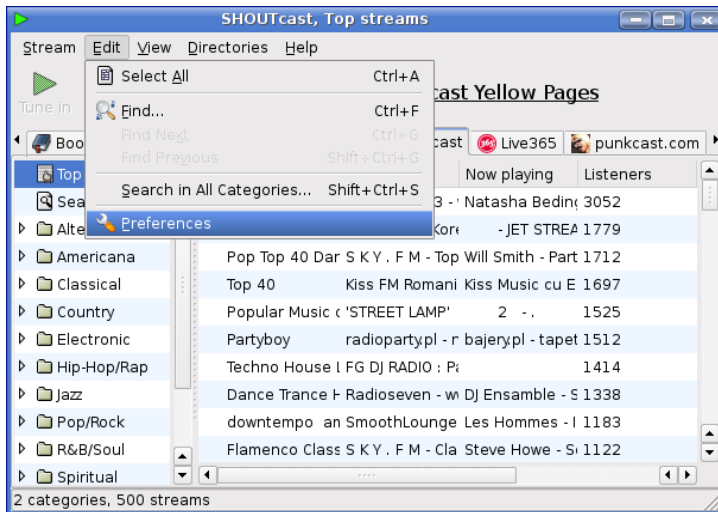


圖 9.37 Streamtuner Preferences

再來我們將 xmms 的字樣改成 beep-media-player 就可以了。

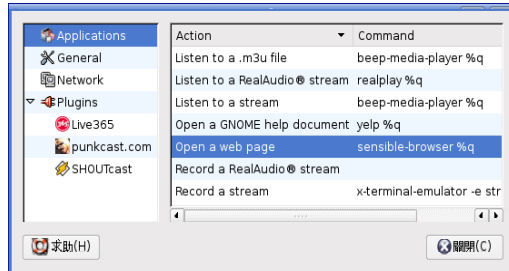


圖 9.38 設定 Streamtuner 播放為 beep-media-player

這樣子就差不多就可以用了，其他的功能就請 user 自行發揮吧！

9.11 用 Linux 看數位電視

這個小節是由企鵝狂熱心提供，在這裡筆者非常感謝他在百忙之中，抽空整理他的安裝筆記 ^_^。

數位電視訊號是將電視影像聲音經過編碼壓縮，加入偵錯編碼之後發送出來的訊號。與一般傳統類比訊號相比，傳統類比訊號容易受到天氣，天線周圍物體移動等等的干擾而產生抖動，雜點等等的雜訊影響畫面品質。數位電視因為加上了錯誤偵測碼，只要訊號來源不要太差，都可以看到與 DVD 畫質媲美的畫面喔。

台灣的五家無線電視台(台視，中視，華視，民視，公視)都已經有發送數位電視訊號，每個電視台都有發送三個不同的節目頻道，所以總共可以看到十五個免費頻道喔。詳細的節目表，以及更多關於數位電視的資訊，可以在台灣數位電視協會的網站上面找到(<http://www.dtv.org.tw/>)。


這個章節要教大家如何在 Linux 上面安裝數位電視的驅動程式，以及使用 Xine 來收看數位電視。

事前準備

安裝 kernel header 與 source – DVB 的驅動程式有許多都還在 linuxtv.org 上面開發而尚未進入正式的 kernel，為了要能編譯 DVB 驅動，需要先裝好目前 kernel 相關的 source。

```
$ sudo apt-get install linux-image-2.6.17-10-generic linux-headers-2.6.17-10
linux-headers-2.6.17-10-generic linux-headers-generic linux-image-generic
```

安裝好之後要用新的 kernel 重新開機，也就是開機時候要 GRUB 選 2.6.17-10-generic。

 安裝 Mercurial 工具 – 這是用來取得 linuxtv.org 上面的程式碼的工具。


```
$ sudo apt-get install mercurial
```

 安裝 gcc,make,automake – 用來編譯原始碼的工具。

```
$ sudo apt-get install gcc-4.1 make automake1.9
```

 下載 DVB 驅動程式與軟體

```
mkdir ~/dvb ; cd ~/dvb
hg clone http://linuxtv.org/hg/v4l-dvb
hg clone http://linuxtv.org/hg/dvb-apps
```

 編譯 DVB

```
cd ~/dvb/v4l-dvb
make;make install
```

這樣會編譯安裝所有的 module

 下載 Firmware (韌體)

如果你的 DVB 裝置是 USB 介面的，通常還需要下載 Firmware。Firmware 是用來驅動硬體的程式碼，通常是放在 FlashROM 裡面。不過現在廠商為了成本以及產品大小的原因，把韌體作成可以隨時載入的。讀者可以在 http://www.linuxtv.org/wiki/index.php/DVB_USB 找到常見的 USB 裝置驅動程式，以及所需要的韌體。

韌體下載網站

```
* http://thadathil.net:8000/dvb/fw/
* http://linuxtv.org/downloads/firmware/
```

例如我的 Compro U500 電視棒使用 DibCom 7700 晶片，因此需要下載 dvb-usb-dib0700-01.fw 這個檔案，下載之後放到 /lib/firmware 裡面，稍後驅動程式載入後，會自動搜尋並且載入韌體。

載入驅動

讓 script 自動幫你載入所有編譯好的驅動

```
cd ~/dvb/v4l-dvb
make load
```

之後打 dmesg 應該就會看到找到電視卡的訊息了

```
Oct 23 22:49:40 linux kernel: usb 2-3: new high speed USB device using ehci_hcd
and address 5
Oct 23 22:49:40 linux kernel: usb 2-3: configuration #1 chosen from 1 choice
Oct 23 22:49:40 linux kernel: dvb-usb: found a 'Compro Videomate U500' in cold
state, will try to load a firmware
Oct 23 22:49:40 linux kernel: dvb-usb: downloading firmware from file
'dvb-usb-dib0700-01.fw'x20
Oct 23 22:49:41 linux kernel: dib0700: firmware started successfully.
Oct 23 22:49:41 linux kernel: dvb-usb: found a 'Compro Videomate U500' in warm
state.
Oct 23 22:49:41 linux kernel: dvb-usb: will pass the complete MPEG2 transport
stream to the software demuxer.
Oct 23 22:49:41 linux kernel: DVB: registering new adapter (Compro Videomate
U500) .
Oct 23 22:49:41 linux kernel: DVB: registering frontend 0 (DiBcom
7000MA/MB/PA/PB/MC) ...
Oct 23 22:49:41 linux kernel: MT2060: successfully identified (IF1 = 1220)
Oct 23 22:49:42 linux kernel: dvb-usb: Compro Videomate U500 successfully
initialized and connected.
Oct 23 22:49:42 linux usbmgr[2980]: vendor:0x185b product:0x1e78
Oct 23 22:49:42 linux usbmgr[2980]: class:0xff subclass:0x0 protocol:0x0
```

dmesg 列出找到 DVB 晶片 DiBcom 7000MA/MB/PA/PB/MC, tuner 晶片 MT2060 的訊息。

✎ 安裝頻道表

```
cd ~/dvb/dvb-apps/
make
```

全台灣各地的頻率可以在這裡找到 <http://ksc91u.googlepages.com/dtv.html>，筆者也建立了嘉義的頻道表放在 <http://ksc91u.googlepages.com/tw.txt>，讀者可以下載來之後修改頻率 Freq 的部份即可，例如 533 就在後面加六個 0 成為 533000000 即可。把修改過的 tw.txt 存在 /tmp/tw.txt。

接下來我們還要用程式掃描節目表，這樣才能讓 Xine 使用。

```
cd ~/dvb/dvb-apps/util/
./scan /tmp/tw.txt >~/xine/channels.conf
```

因為掃描出來的頻道表沒有辦法顯示中文，反而是一堆亂碼，我們可以手動編輯 ~/xine/channels.conf 這個檔案，把每一行第一個冒號：前面的頻道名稱修改成我們看得懂的代號，你也可以使用中文，但是只有在 gxine 裡面看的到，因為 xine 目前還不能正確顯示中文。修改之後過的檔案長得像這樣 <http://ksc91u.googlepages.com/channels.conf>。頻道名稱是 [xxxx] 的似乎都是空白沒有辦法觀看的，可以整行刪掉。



圖 9.39 gxine 播放畫面

使用 gxine 時候，可以用滑鼠滾輪來選擇上下頻道，選好要看的頻道之後按左鍵切換。也可以用 Page Up/Down 來上下轉台。

以上是個人在 Linux 上面安裝數位電視的過程與讀者分享。當然 Linux 尚有如 Kaffeine 或是 Mplayer 等軟體可以播放數位電視，Kaffeine 可以顯示每日的節目時刻表，不過也一樣是亂碼，還要等待有心人士的修正。Mplayer 個人則覺得沒有 Xine/Gxine 那樣方便的介面可以使用。如果在安裝過程中遇到甚麼問題，可以到 LinuxTV 的 Wiki (http://www.linuxtv.org/wiki/index.php/Main_Page) 上面查詢自己的電視卡類型。或是也歡迎將各地收訊的狀況回報與大家分享討論喔。

CHAPTER 10

一些實用的應用軟體

看過了前面介紹的網路應用程式、影音多媒體，是不是還覺得意猶未盡？Linux 上有圖片瀏覽器嗎？我要怎麼燒 CD/DVD 呢？還有怎麼做簡報、報告。Linux 上的這些軟體怎麼裝怎麼用，要不要錢呢？答案是一樣不用錢哦！而且也是一樣 apt-get 就可以裝好囉！使用的方式也很簡單，就請讀者不厭其煩的接著看吧！

10.1 圖片瀏覽與編輯程式

在 Linux 上有許多的圖片瀏覽程式，在下面筆書會介紹幾個簡單易用的圖片瀏覽程式，還有內建的圖片編輯程式 (Gimp)，Gimp 是個很好用的圖片編輯程式，有如 windows 下的 photoshop，筆者在寫這本書時也是用 Gimp 修圖哦！接下來就請看看這些程式的基本用法，與安裝方法。

10.1.1 Linux 中的 photoshop—Gimp

Gimp 是個簡單易用的圖片編輯程式，也是 Ubuntu 內建的，Gimp 的使用者介面很簡單好用，筆者在寫這本書也是用 Gimp 來修圖，不過因為不是要寫 Gimp 教學，所以就簡單的介紹一下使用介面，和筆者修圖時的做法。

✎ 1. 安裝 Gimp

萬一你的 Ubuntu 裡沒有 Gimp，請使用以下指令安裝。

```
$ sudo apt-get install gimp
```

✎ 2. 使用介面簡介

Gimp 的開啟中畫面，剛開啟時 Gimp 會載入程式以及外掛。



圖 10.1 Gimp 讀取程式中

接著開啟後會有小提示，如果不想看可以先關掉。

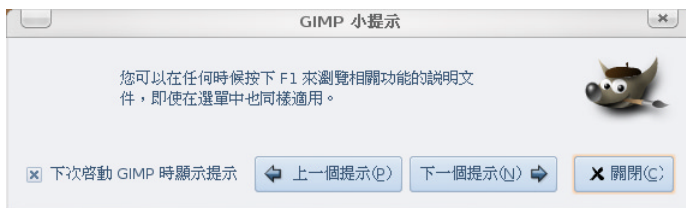


圖 10.2 Gimp 小提示

Gimp 的工具列，基本上所有的功能都可以在這裡找到，包括：開啟檔案、影像的縮放、所有的工具小圖示、畫筆顏色、筆刷大小、操作步驟等等。



圖 10.3 Gimp 工具列

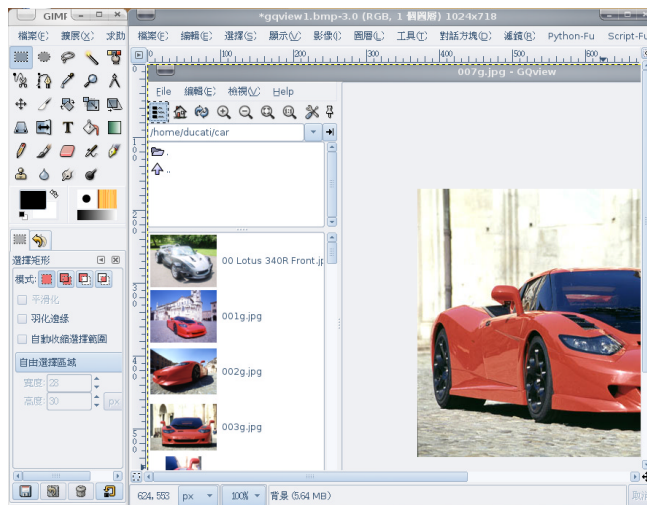


圖 10.4 Gimp 開啟圖片

✎ 3. 加入回復步驟

這樣可以讓你做錯的地方重來

增加功能選單，按下小方塊選 加入分頁--> 復原紀錄



圖 10.5 加入復原紀錄

然後在左邊的工具列就會出現復原紀錄，你就可以選擇想回復的步驟了。



圖 10.6 選擇復原紀錄

✎ 4. 簡單的使用範例

將選擇範圍描邊，在書本上的圖片，可以看到很多小框框的標示，筆者在這裡簡單的說明步驟。

先選好你要描邊的顏色，先點選前景顏色的地方。

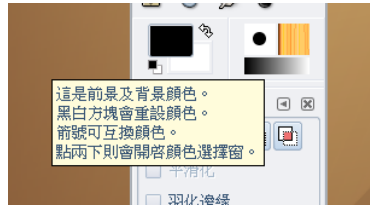


圖 10.7 選擇前景顏色

會出現選擇顏色的視窗，選擇你想要的顏色。



圖 10.8 顏色選擇視窗

然後前景顏色就會變成紅色了。

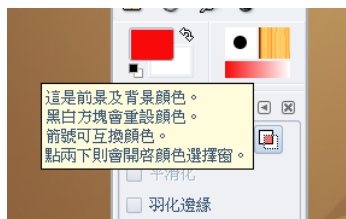


圖 10.9 前景顏色變成紅色

接著用選擇矩形區域工具，選好你想描邊的地方。



圖 10.10 使用矩形區域工具

點選 編輯--> 依選擇區域描邊。



圖 10.11 依選擇區域描邊

然後會出現設定視窗，如果不想描邊的線條太粗，可以設 3.0。



圖 10.12 設定描邊粗細

這樣子就可以描出細細的紅邊了。

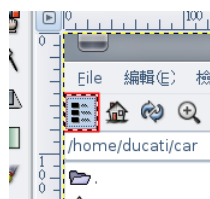


圖 10.13 描邊完成

✎ 5. 其他的功能選項



圖 10.14 選擇的功能選項

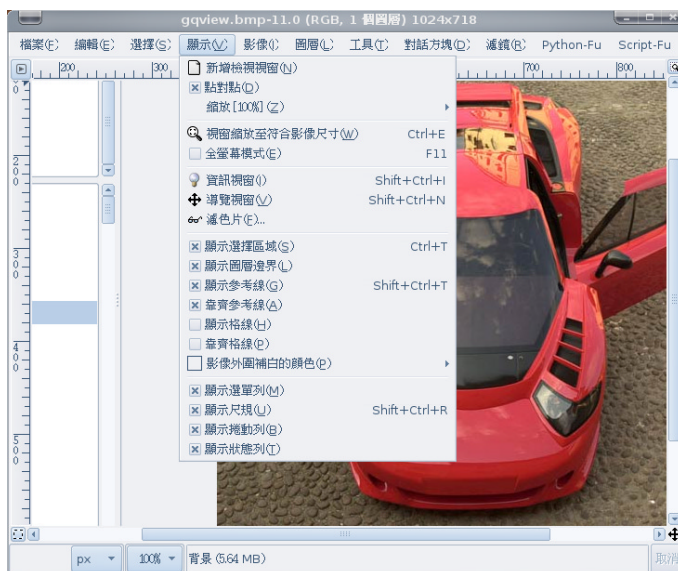


圖 10.15 顯示的功能選項

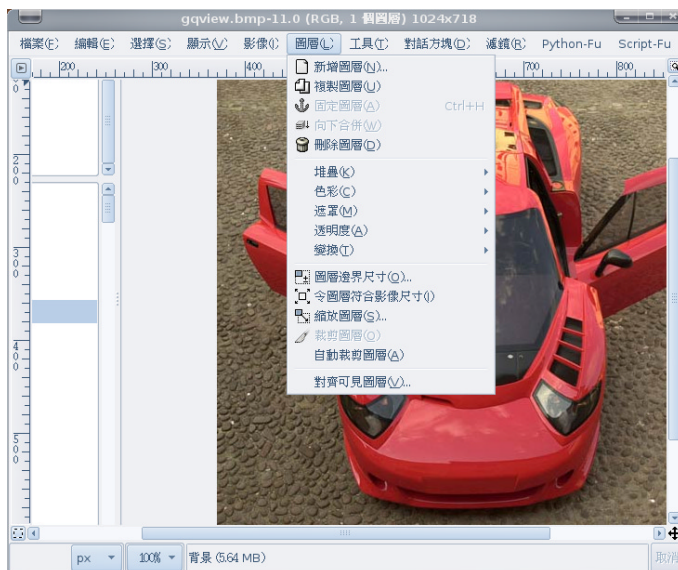


圖 10.16 圖層的功能選項

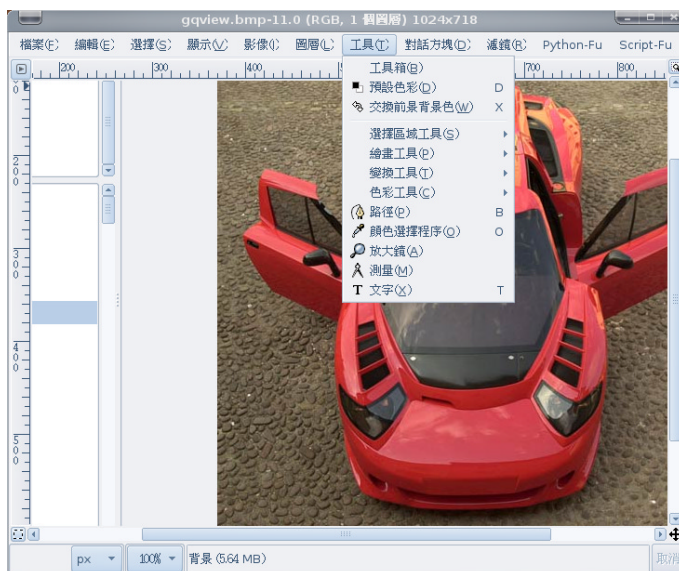


圖 10.17 工具的功能選項

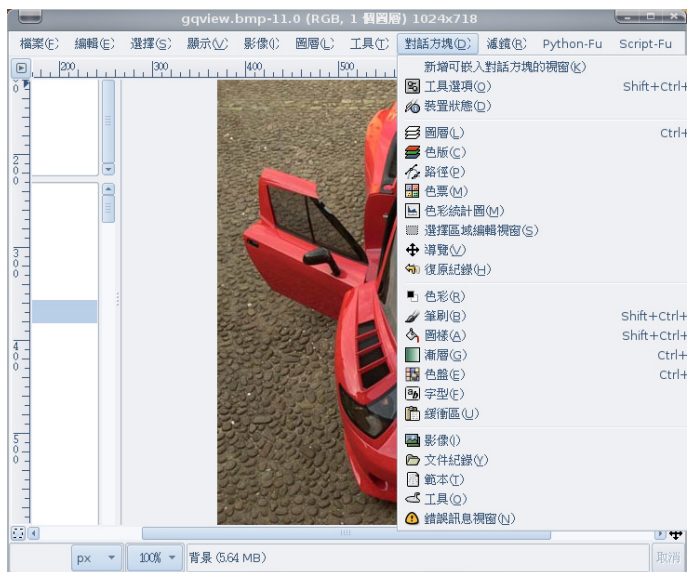


圖 10.18 對話方塊的功能選項

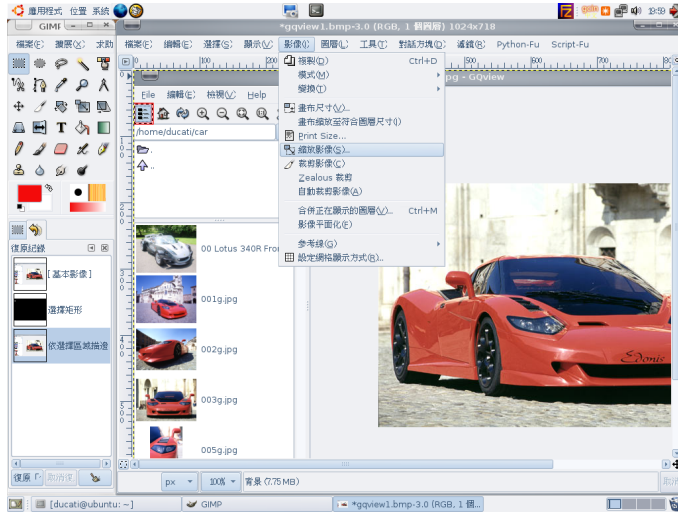


圖 10.19 影像的功能選項，可以用來縮放影像的大小哦！

10.1.2 輕便快速的圖形瀏覽程式－GQview

🔧 安裝 Gqview

```
$ sudo apt-get install gqview
```

啟動 Gqview，應用程式--> 美工繪圖--> Gqview

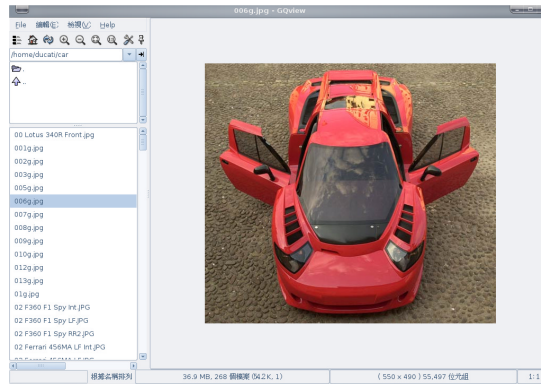


圖 10.20 Gqview

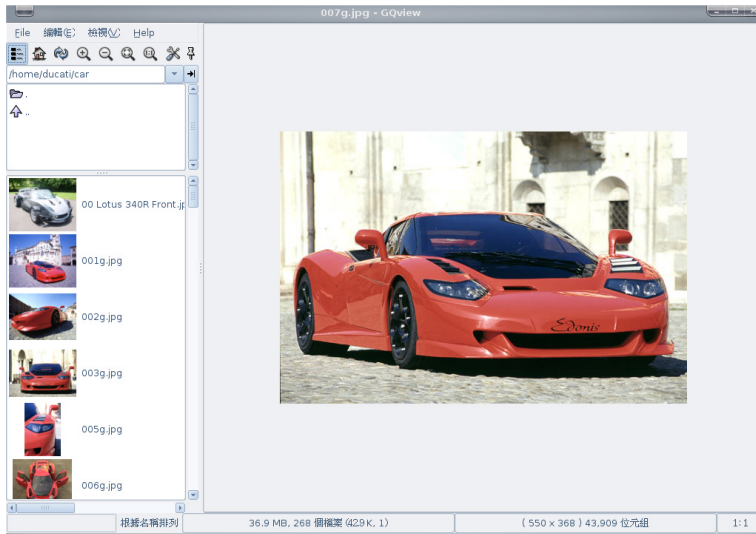


圖 10.21 圖片列表顯示



圖 10.22 全螢幕模式，可以按 F 切換

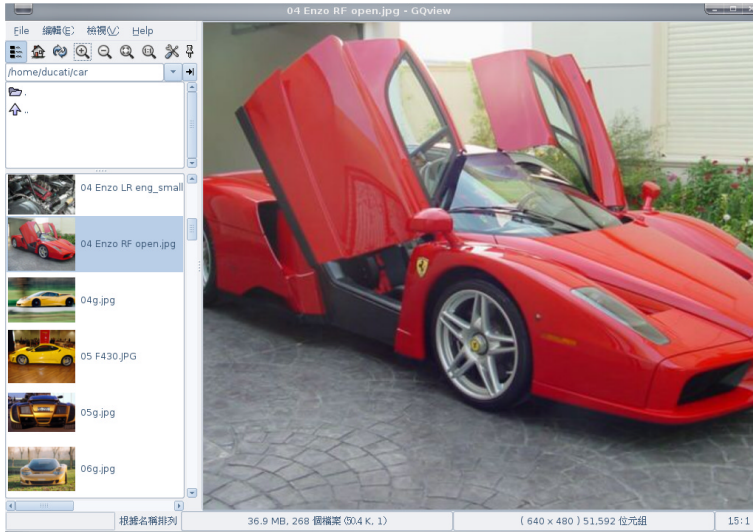


圖 10.23 放大顯示

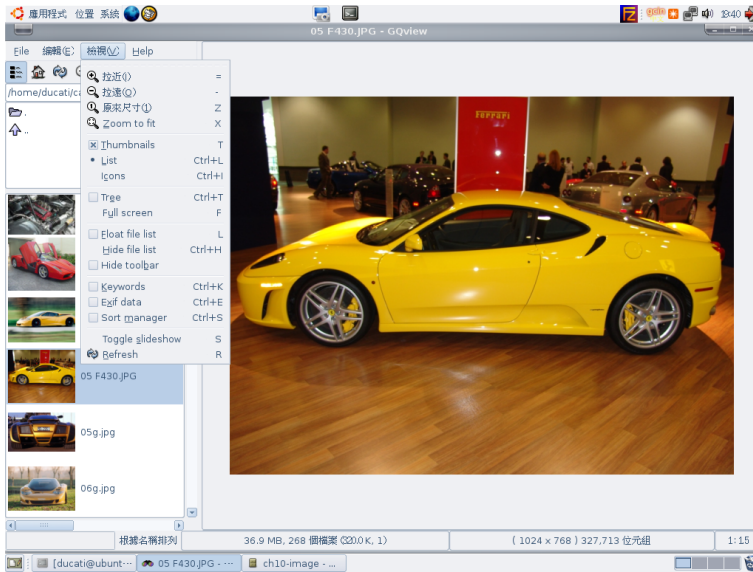


圖 10.24 圖片檢視功能選單

10.1.3 簡易的影像檢視器－Gwenview

Gwenview 的顯示方式是，若你在已有圖片的資料夾裡，所有的圖會變成小縮圖顯示，被點選到的圖，則會在旁邊顯示大一點點的圖，不過 Gwenview 還可以加上外掛（plugin），加上 plugin 後可以將你的圖檔製成影片檔，或是燒錄成 CD/DVD。

🔧 安裝 Gwenview

```
$ sudo apt-get install gwenview
```

開啟 Gwenview，應用程式--> 美工繪圖--> Gwenview

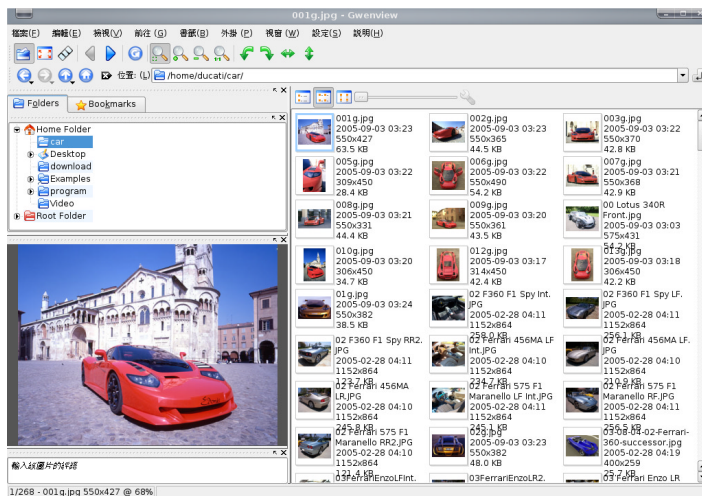


圖 10.25 Gwenview

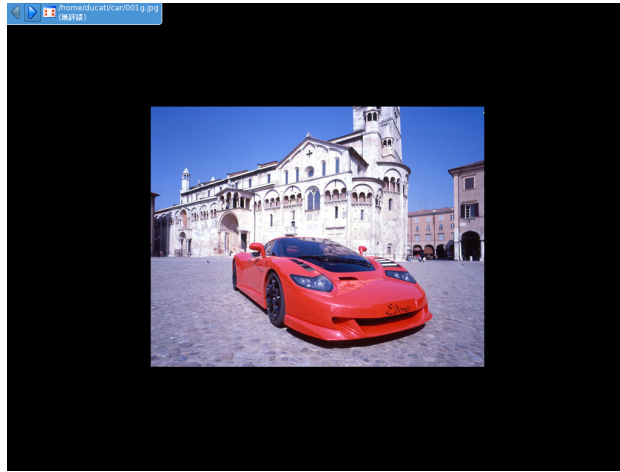


圖 10.26 全螢幕模式，可按 F 變換

✎ 裝上 Gwenview pluing，讓 Gwenview 可以將影像燒成光碟

```
$ sudo apt-get install kipi-plugins
```

然後可以在 外掛--> 匯出--> Archive to CD/DVD，匯出成光碟

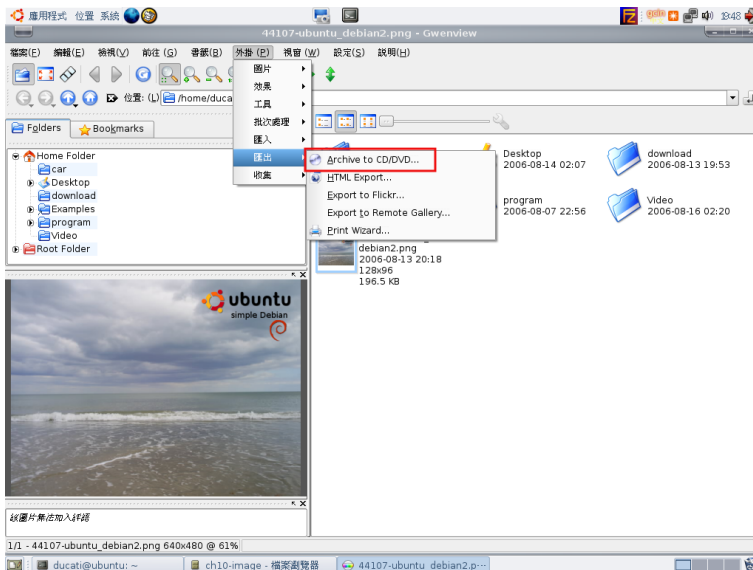


圖 10.27 匯出成 CD/DVD

也可以匯出成 MPEG 影片檔，外掛--> 工具--> Create MPEG Slide Show

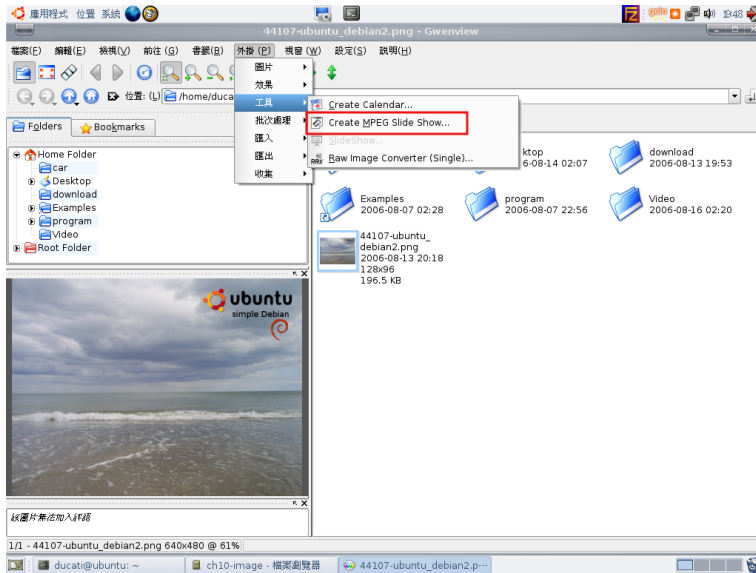


圖 10.28 匯出成 MPEG 檔

10.1.4 好用的圖片管理程式－gThumb

📁 安裝 gThumb

```
$ sudo apt-get install gthumb
```

開啟 gThumb，應用程式--> 美工繪圖--> gThumb 圖片管理工具

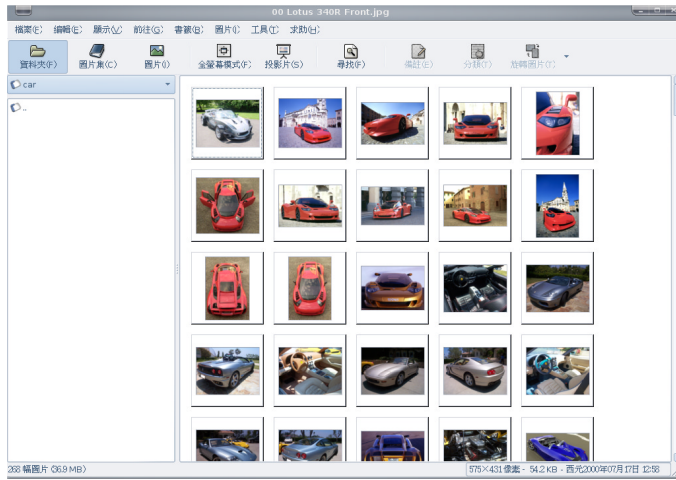


圖 10.29 gThumb

gThumb 的全螢幕模式

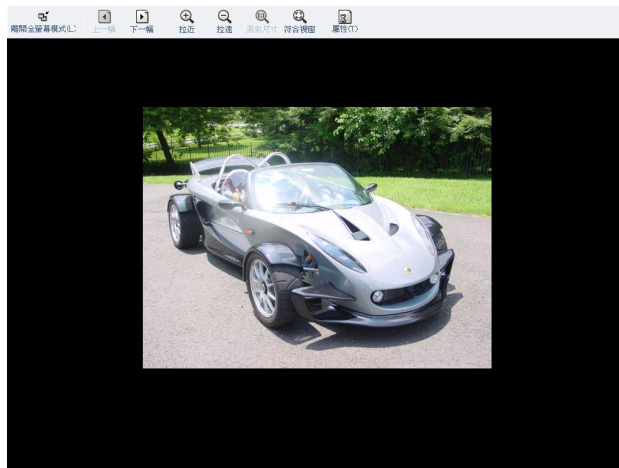


圖 10.30 全螢幕模式

用 gThumb 產生索引圖，縮放旋轉圖片、更改日期



圖 10.31 gThumb 工具

gThumb 的 slide 模式

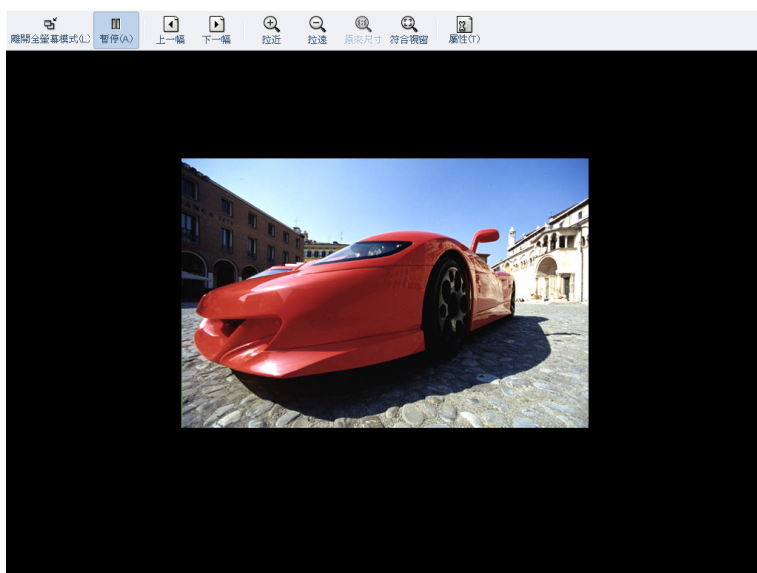


圖 10.32 slide 模式

10.1.5 預設的圖片管理程式—Eye of GNOME

開啟 Eye of GNOME (eog) 的方式有兩種，第一種是圖片上點兩下，第二種是在圖上按右鍵，選擇以“圖片瀏覽程式”開啟



圖 10.33 開啟 Eye of GNOME

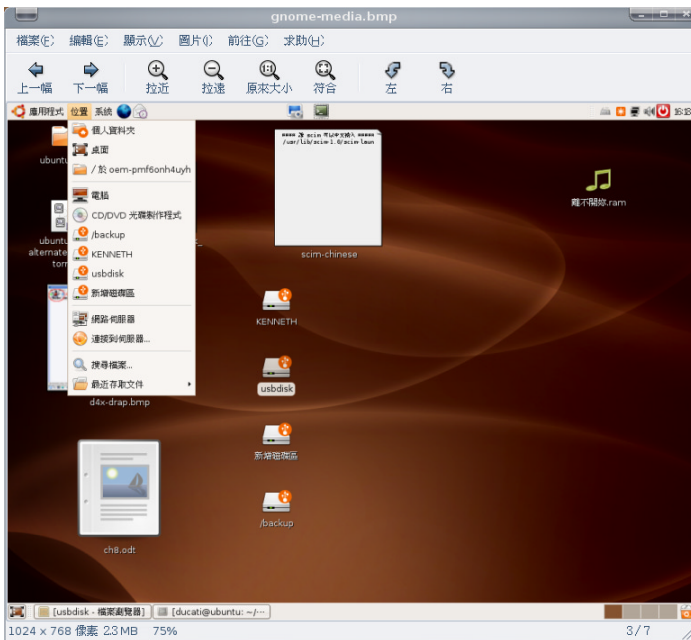


圖 10.34 Eye of GNOME

10.2 StarDict (星際譯王) – 好用的翻譯軟體

✎ 1. 安裝 StarDict

```
$ sudo apt-get install stardict
```

✎ 2. 開啟 StarDict，應用程式--> 附屬應用程式--> 星際譯王

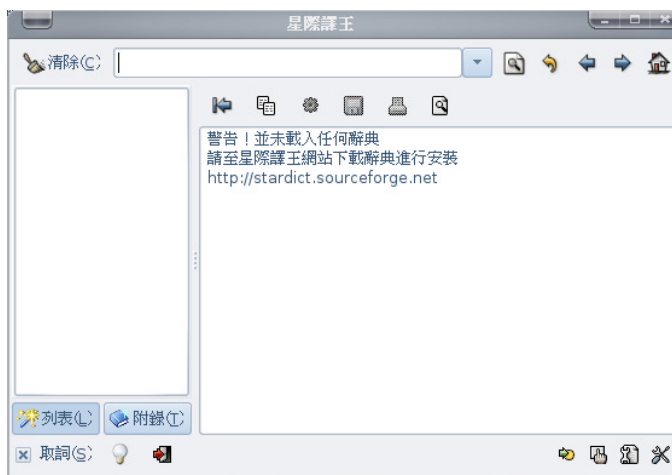


圖 10.35 StarDict

10.2.1 安裝不同的字典檔

到 StarDict 網站下載字典檔案：

```
http://stardict.sourceforge.net/Dictionaries\_zh\_TW.php
```

選擇你想裝的字典檔，記得下載**英漢字典**和**漢英字典**，一起安裝才有用。

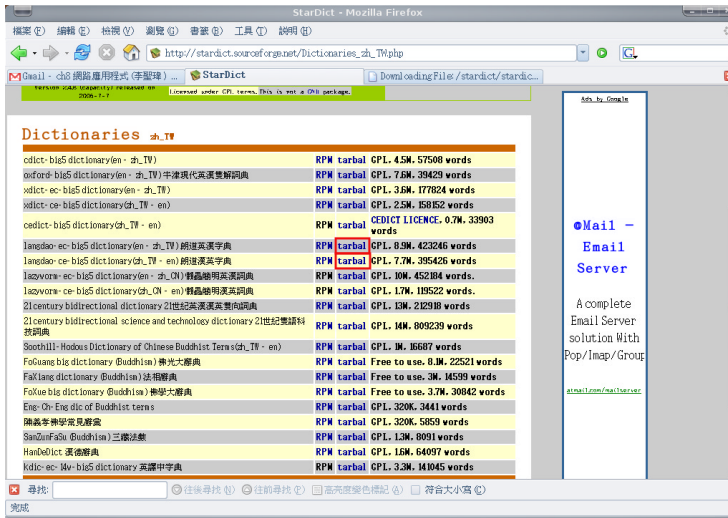


圖 10.36 下載字典檔

下載回來後在 /usr/share/stardict/dict 解壓縮，兩個檔案都要哦。

```
$ sudo cp /字典檔所在目錄/字典檔.tar.bz2 /usr/share/stardict/dict/
```

解壓縮，一樣兩個檔案都要。

```
$ cd /usr/share/stardict/dict/ ; sudo tar jxvf 字典檔.tar.bz2
```

接著開啟 StarDict，然後選擇右下角的辭典管理，可以看到英漢字典和漢英字典，都安裝好了。

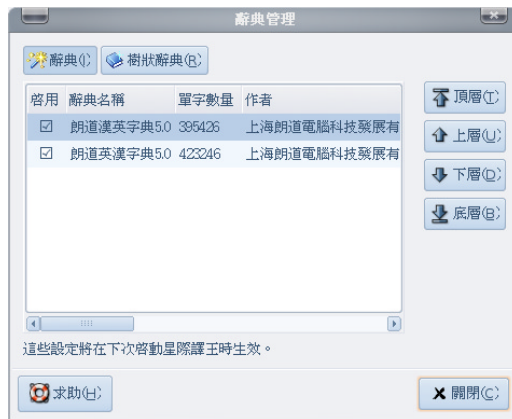


圖 10.37 啟用字典檔

然後輸入英文字看看，是否有翻譯得出來。若不行的話將 StarDict 重開。

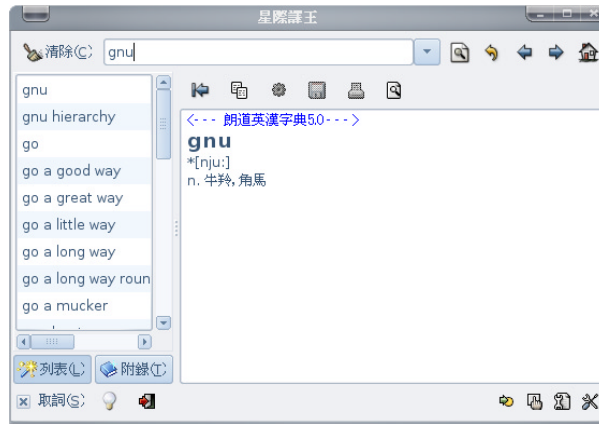


圖 10.38 測試字典

StarDict 有**取詞**的功能，也就是說只要在文字上用滑鼠選取，就會出現即時翻譯。



圖 10.39 取詞

10.2.2 安裝人聲發音檔

先下載約 80 MB 的發音檔

```
http://prdownloads.sourceforge.net/stardict/WyabdcRealPeopleTTS.tar.bz2?download
```

下載回來後解開，然後將解開的資料夾 copy 到 /usr/share/

```
$ tar jxvf WyabdcRealPeopleTTS.tar.bz2 ; sudo mv WaybdcRealPeopleTTS/ /usr/share
```

接著重開 Stardict，開啟**偏好設定**，然後在**浮動視窗**--> **選項**--> **彈現時單字發音**，打勾即可。



圖 10.40 取詞時自動發音

不過並不是每個單字都有發聲哦！但是常用的單字基本上都有發音。

10.3 燒錄軟體

10.3.1 K3b—KDE 底下的燒錄軟體

K3b 在 Ubuntu 下預設就安裝好囉！如果你沒有 K3b 的話，請用以下指令安裝。

```
$ sudo apt-get install K3b
```

可在**應用程式**--> **影音**--> **K3b** 開啟，第一次開啟 K3b 時，K3b 會偵測你的硬體，以調整你的燒錄速度，如果你有自己的選擇，可在這裡更改，或是在 K3b 開啟後在燒錄時設定。

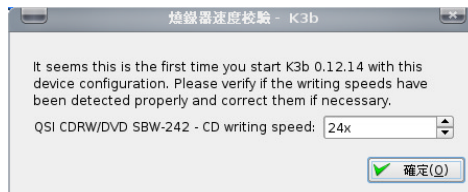


圖 10.41 設定燒錄速度

可以在下面藍色圖裡，選擇你想燒錄的光碟型式，基本上有：音樂光碟、資料光碟、資料 DVD、複製光碟、燒錄 CD 映像檔、燒錄 DVD 映像檔。



圖 10.42 選擇想燒錄的光碟及檔案型式

選好燒錄型式後，只要打開你的資料夾，在檔案上點兩下即可加入燒錄。

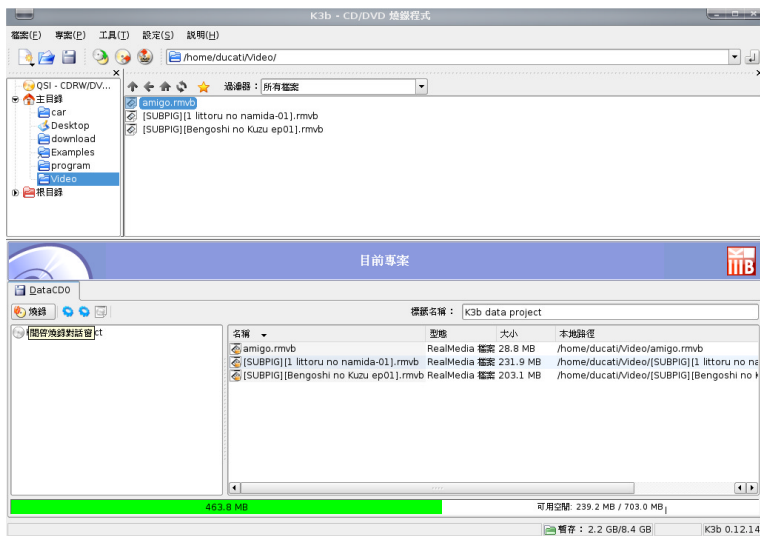


圖 10.43 加入檔案

燒錄設定及開始燒錄



圖 10.44 燒錄

10.3.2 GnomeBaker – Gnome 底下新的燒錄軟體

✎ 安裝 GnomeBaker

```
$sudo apt-get install gnomebaker
```

可在 **應用程式**--> **影音**--> **GnomeBaker** 開啟，使用方式更簡單，在 GnomeBaker 的工具列上，就有許多的燒錄型式可以選擇，燒錄光碟的型式，跟 K3b 差不多。

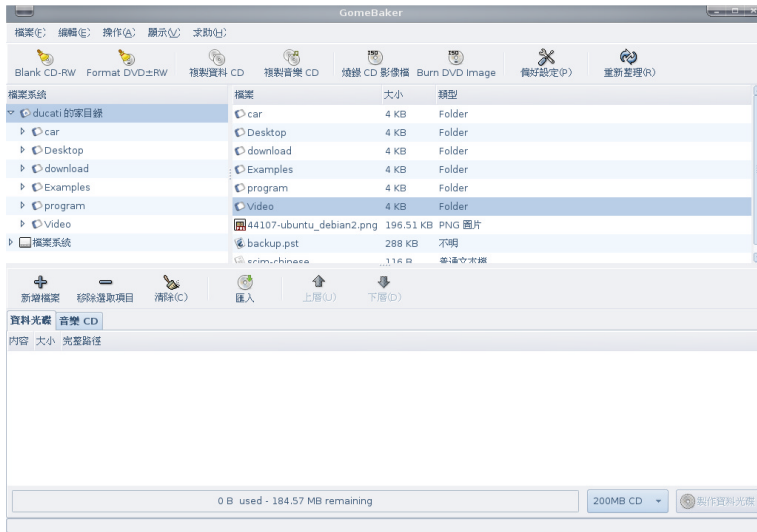


圖 10.45 GnomeBaker

10.4 文書處理軟體

10.4.1 文件製作軟體—OpenOffice.org Writer

筆者在寫書時也會用 OpenOffice.org Writer 來做文書處理。以下是 OpenOffice.org Writer 的外觀。

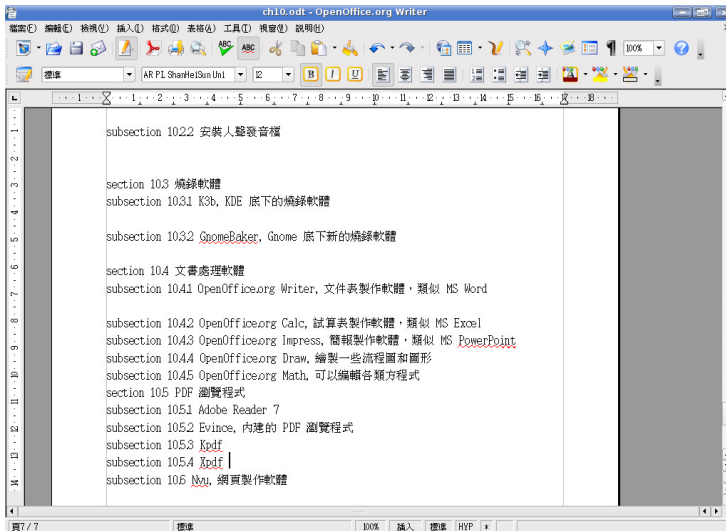


圖 10.46 OpenOffice.org Writer

10.4.2 試算表製作軟體—OpenOffice.org Calc

跟 MS 的 Excel 很像，而且功能上也沒有打折扣哦！

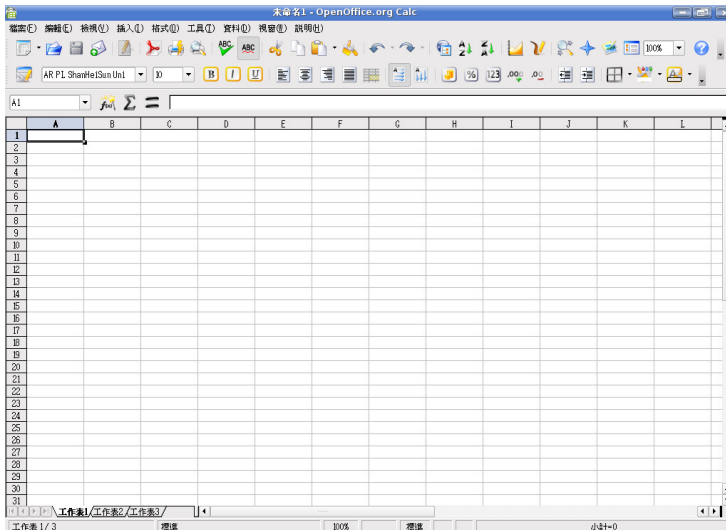


圖 10.47 Openoffice.org Calc

10.4.3 簡報製作軟體－OpenOffice.org Impress

簡單好用的簡報軟體，從 MS PowerPoint 轉換過來，一點都沒有問題，再加上 Xgl 的話，報告時旋轉你的桌面，是不是很棒呢？

一開始也有簡報精靈，引導製作簡報。選擇要開啟舊檔或是新檔。

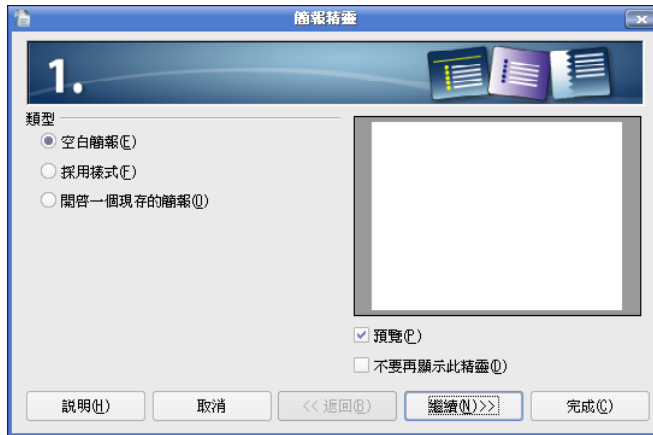


圖 10.48 開新檔案

選擇簡報樣式

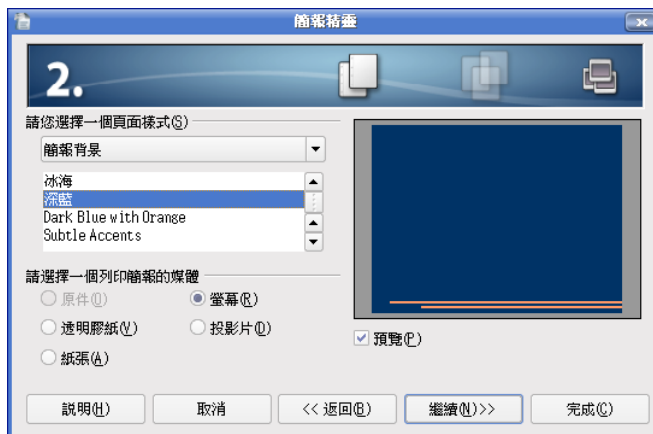


圖 10.49 選擇簡報樣式

選擇投影片變換方式



圖 10.50 投影片變換方式

開啟製作簡報檔囉！

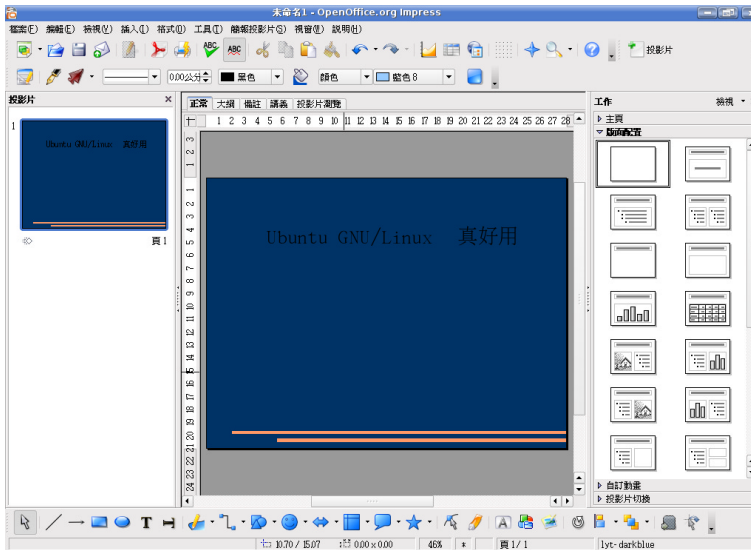


圖 10.51 OpenOffice.org Impress

10.4.4 流程圖與圖形繪製 – OpenOffice.org Draw

```
$ sudo apt-get install openoffice.org2-draw
```

開啟 OpenOffice.org Draw 時，你可以先開啟 OpenOffice 主程式，在 console 下輸入 `openoffice &`。

```
$ openoffice &
```

然後在開新檔案圖式旁的三角型按一下，選繪圖就可以開啟 OpenOffice.org Draw。

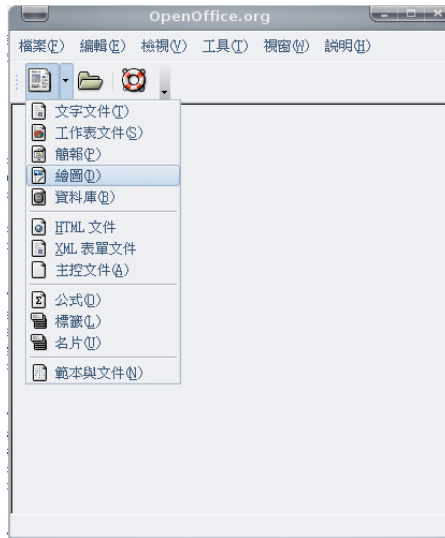


圖 10.52 開啟 Draw

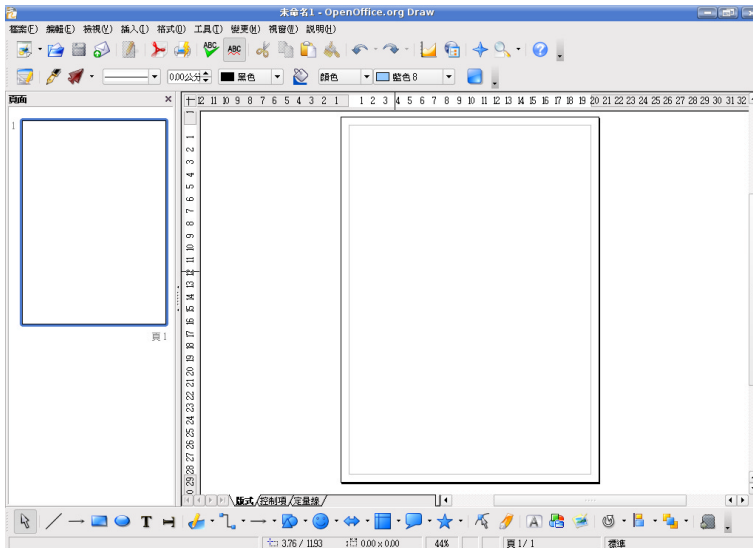


圖 10.53 OpenOffice.org Draw

10.4.5 方程式編輯 – OpenOffice.org Math

✎ 安裝 OpenOffice.org Math

```
$ sudo apt-get install openoffice.org2-math
```

✎ 開啟 OpenOffice.org Math 時，你可以先開啟 OpenOffice 主程式，在 console 下輸入 `openoffice &`。

```
$ openoffice &
```

然後在開新檔案圖示旁的三角形按一下，選**公式**就可以開啟 OpenOffice.org Math。

10.5 PDF 瀏覽程式

10.5.1 Adobe Reader 7

✎ 安裝 Adobe Reader 7 以及外掛

```
$ sudo apt-get install acroread acroread-plugins cmap-adobe-cns1
```

如果你的 Adobe Reader 顯示不了日文字型，或是其他國家的字型，你可以先 apt-cache search 一下，安裝你所需要的字型。

```
ducati@X20:~$ sudo apt-cache search cmap-adobe
cmap-adobe-cns1 - CMaps for Adobe-CNS1
cmap-adobe-gbl1 - CMaps for Adobe-GB1
cmap-adobe-japan1 - CMaps for Adobe-Japan1
cmap-adobe-japan2 - CMaps for Adobe-Japan2
cmap-adobe-korea1 - CMaps for Adobe-Korea1
```

✎ 開啟 Adobe Reader 7

應用程式--> 辦公--> Adobe Reader，一開始會問你是否同意授權條款，如果同意請選 Accept 以開啟 Adobe Reader。

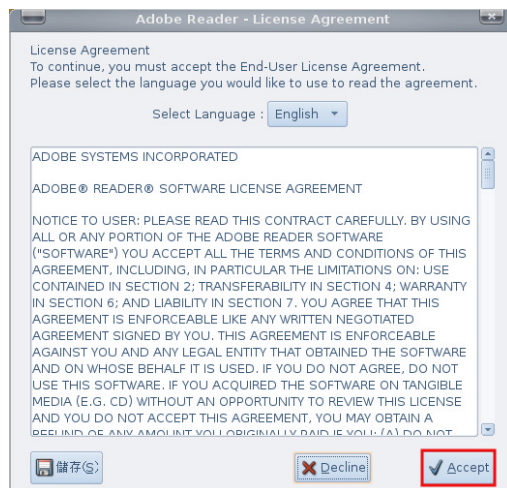


圖 10.56 Adobe Reader – License

Adobe Reader 7

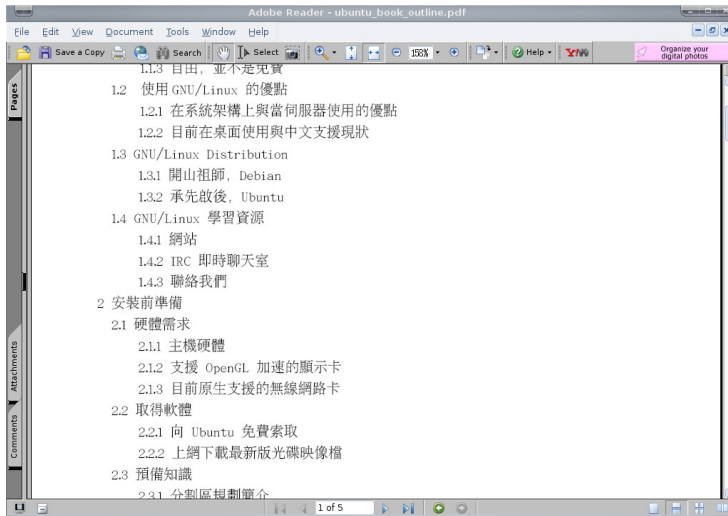


圖 10.57 Adobe Reader 7

10.5.2 Evince — 內建的 PDF 瀏覽程式

開啟 Evince，在 pdf 檔上按右鍵，選以“文件檢視器”開啟。



圖 10.58 開啟 Evince

Evince

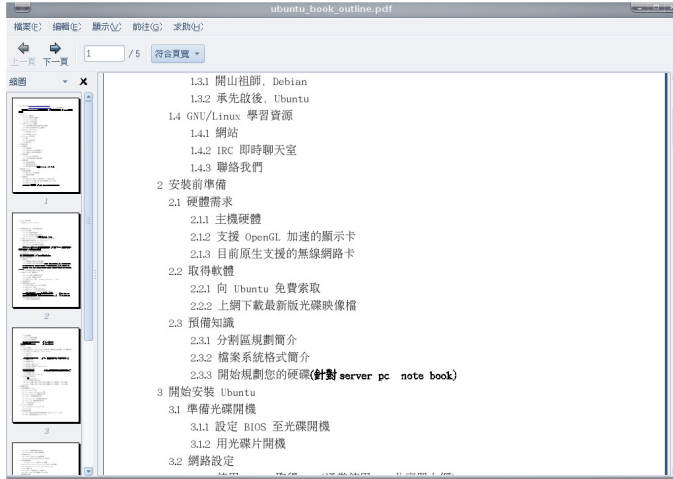


圖 10.59 Evince

10.5.3 Kpdf

安裝 Kpdf

```
$ sudo apt-get install kpdf
```

開啟 Kpdf，應用程式--> 美工繪圖--> Kpdf

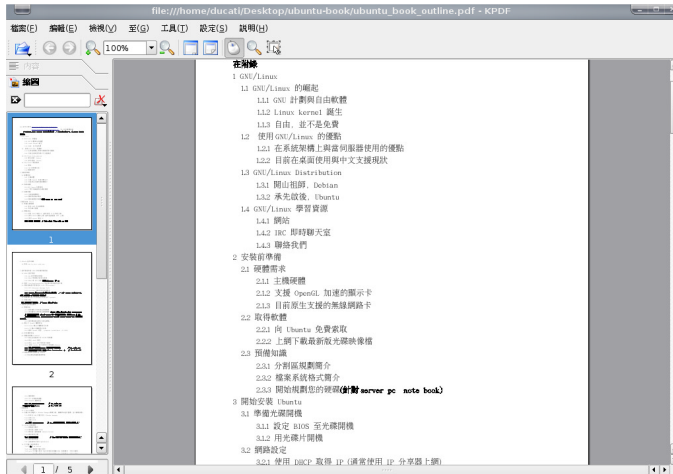


圖 10.60 Kpdf

10.5.4 Xpdf

安裝 Xpdf

```
$ sudo apt-get install xpdf xpdf-chinese-traditional
```

開啟 Xpdf，應用程式--> 美工繪圖--> Xpdf

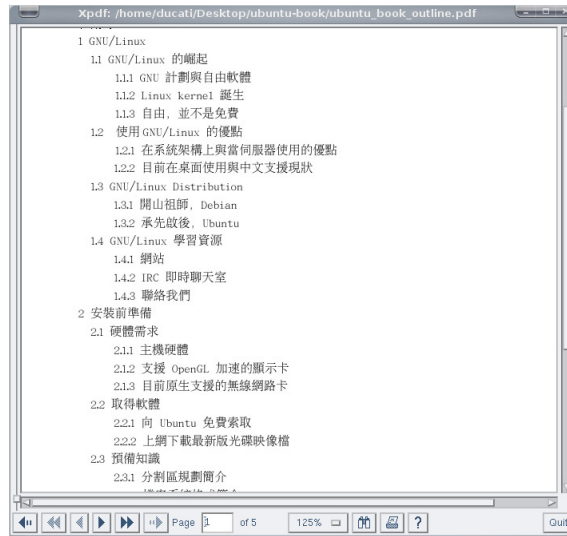


圖 10.61 Xpdf

10.6 網頁製作軟體－Nvu

安裝 Nvu

```
$ sudo apt-get install nvu
```

開啟 Nvu，應用程式--> 軟體開發--> Nvu，一開始 Nvu 會出現一個小視窗，跟你說 Nvu 開發人員想知道你有下載 Nvu，只要按下 OK 就會送出一個 ping 給他們團隊的電腦上，在這裡按 OK 就可以囉！

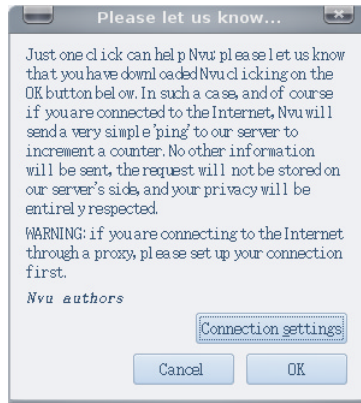


圖 10.62 Ping Nvu Team

Nvu

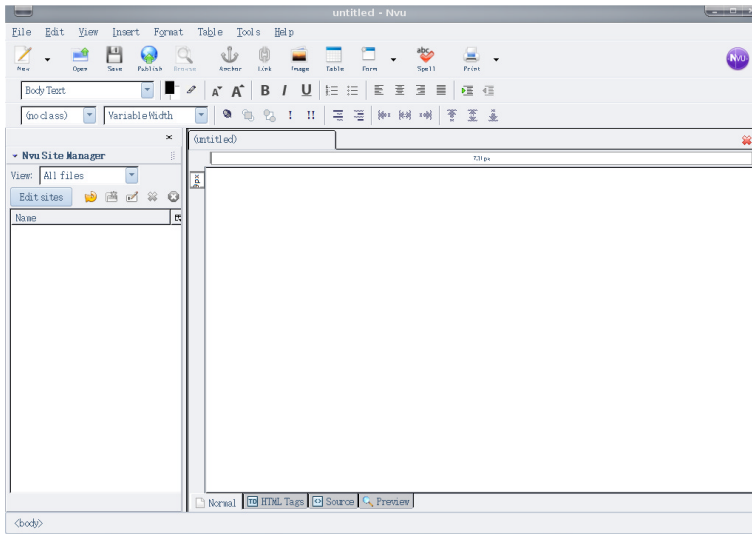


圖 10.63 Nvu

10.7 PCMan File Manager

這是 PCMan 大哥寫的 File Manaer，相當輕便好用，可說是麻雀雖小五臟俱全，執行速度快。讓我們來看看 PCMan File Manager 有什麼特點：

- 輕量化與快速
- 可以裝在舊機器上
- 分頁瀏覽（如 Firefox 一樣）
- 托曳與放入的支援
- 檔案可以在分頁中托曳
- 縮短讀取大量資料夾的時間
- 檔案關聯性支援
- 書籤功能
- 磁碟檔案編碼的轉換
- 提供圖示與目錄樹的瀏覽方式
- 依照 FreeDesktop.org 所寫出來的
- 簡潔而有親和力的介面（GTK+ 2）

官方網址 <http://pcmanfm.sourceforge.net/>

安裝方式

在 `/etc/apt/sources.list` 加上

```
deb http://cle.linux.org.tw/candyz/Ubuntu/edgy i386/
```

然後加上 APT Key, 在終端機下輸入

```
$ wget http://cle.linux.org.tw/candyz/Ubuntu/candyz.key -O - | sudo apt-key add -
```

接著 `update` 並 `install`

```
$ sudo apt-get update
$ sudo apt-get install pcmanfm
```

執行 PCManFM

安裝後可以在**應用程式-->系統工具-->PCMan 檔案管理程式**找到執行。

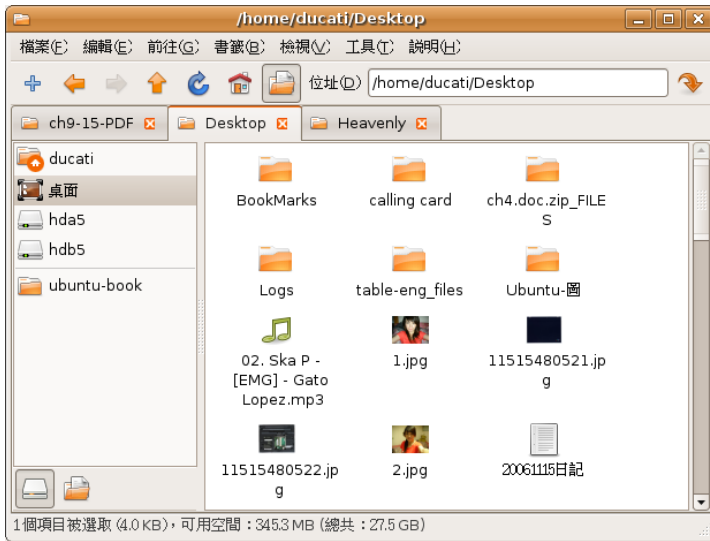


圖 10.64 PCMan File Manager

10.8 Lightweight X11 Desktop Environment

輕巧的 Linux desktop—LXDE (Lightweight X11 Desktop Environment)，是由 pcman 與 jserv 合作的專案，藉由組裝一些輕巧的程式，提供桌面新選擇。預先裝好了中文介面，有新酷音輸入法，有 PCManX BBS 連線程式等等。安裝方式與圖片可以參考：

用 QEMU 模擬的方式：

http://freesf.tw/modules/newbb/viewtopic.php?viewmode=flat&topic_id=4863&forum=30

若你已經安裝好 Ubuntu、Debian，或是其他 Linux 版本也可以參考這裡安裝：

<http://lxde.sourceforge.net/install.html>

下圖是筆者試裝 LXDE 的樣子，裝在 IBM X20 (PIII 600 RAM 192 MB) 還真是順呢！

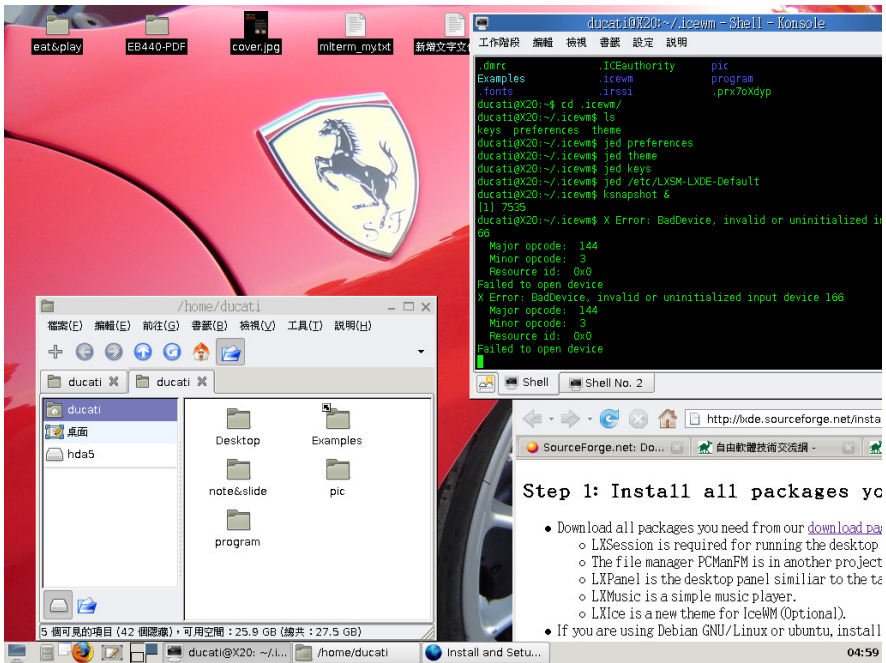


圖 10.65 LXDE

CHAPTER 11

用 Wine 來執行 Windows 的 遊戲與程式

11.1 wine 概觀

Wine 這個計畫是在 1993 年所開始的，由 Bob Amstadt 及 Eric Youngdale 發起的，當初是為了在 Linux 上可以執行 Windows 3.1 的程式。

要執行 Windows 的程式必需要有 DLL 檔才行，DLL 檔是 Windows 的函式庫，但是 Windows 的 DLL 檔多半沒有完全公開，更別提有原始程式碼作參考，所以支持 Wine 的程式設計師是用逆向工程分析並重寫程式，這是很難的工作，而且一定會有許多錯誤和 bug，Wine 為了要追 Windows 上的程式，所以會常常出新的版本，Wine 又可以稱為「Wine Is Not an Emulator」，Wine 不是模擬器。不過話說回來，經小弟測試過的感覺，用 Wine 來跑以 DirectX 所寫的遊戲時，感覺比在 Windows 上執行的效能還要好。有人說過 Wine 就像是一塊黏土，要親手將 Wine 捏成你喜歡的樣子，不過 Wine 其實不難，只是網路上很少有完整的教學文章，都是很散亂的。在這裡讓筆者介紹一下，IE 與 WoW、Steam 的安裝筆記吧！

- Wine 官方網站 (<http://www.winehq.org/>)：你可以到這裡看看 Wine 的詳細介紹，以及 Wine 的新聞。
- Wine 的 .deb 檔下載網址 (<http://wine.budgetdedicated.com/archive/index.html>)：你可以直接 apt-get 安裝，但如果要玩魔獸世界請參考下個小節的 Wine 裝法。
- Wine 可以裝什麼？大部份你在 Windows 上看到的常用軟體，像是 Office、MSN、IE、games 等等的都可以裝，讀者可以上 Wine 的 AppDB 網頁看看，可以在裡面找看看你想裝的 Windows program，裡面的東西真的很多，照著網頁上的 HOWTO 就可以將 Windows program 裝起來。幾乎是在 Windows 上用得的軟體都裝得起來，不管是網頁設計、3D 繪圖、程式設計，等等的軟體都能裝哦！
- Wine AppDB (<http://appdb.winehq.org/>)

11.2 裝上 IE

有個很方便的軟體叫做 IEs 4 Linux，可以讓你很方便快速的裝上 IE，而且大部份 Wine 的版本都適用，IEs 4 Linux 是個自由軟體，使用 GPL 授權，關於 GPL 授權可以看這裡 <http://www.gnu.org/licenses/gpl.txt>，不過請注意 IE 是有版權的軟體，你必需要接受授權條例^❶，Ies 4 Linux 的網頁上有提醒你，使用 IEs 4 Linux 安裝任何微軟的軟體時，你必須是已經購買並持有 Windows 的授權。沒有經過授權的安裝是違法的，而且筆者與 IEs 4 Linux 將不負任何的責任。

11.2.1 IEs 4 Linux

<http://www.tatanka.com.br/ies4linux/index-zh.html>

❶ <http://www.microsoft.com/msdownload/ieplatform/ie/license.txt>

安裝 IEs 4 Linux

先到 <http://wine.budgetdedicated.com/archive/index.html>，下載最新版本的 Wine 安裝，依以下指令（如果你只要安裝 IE 的話，直接 apt-get 安裝 Wine 即可，若想安裝魔獸世界或是 steam 的話，Wine 還要再 patch 過才行，筆者無法保證其他軟體是否也要 patch，因筆者只有裝過 IE、MSN、魔獸世界和 steam），若你想同時安裝 IE 和魔獸世界的話，建議先參考安裝魔獸世界的章節，將 Wine 裝起來。要用 IEs 4 Linux 裝 IE，必需要先裝上 Wine 以及 cabextract。

1. 裝上 Wine

在這裡可以直接 apt-get 安裝，或是到上面給的 .deb 檔下載安裝。

```
$ sudo apt-get install wine
$ sudo dpkg -i wine_VERSION.deb
```

雖然 apt-get 就可以安裝 Wine 了，不過筆者建議還是安裝最新版本的 Wine。

2. 裝上 cabextract

```
$ sudo apt-get install cabextract
```

11.2.2 下載 IEs 4 Linux

<http://www.tatanka.com.br/ies4linux/download.html>

解開 IEs 4 Linux

```
$ tar zxvf ies4linux-VERSION.tar.gz
```

VERSION 指的是你 ies4linux 的版本哦！

3. 安裝 IEs 4 Linux

```
$ cd ies4linux-VERSION/ ; ./ies4linux
```

接著會出現幾個小選項：

```
IE 6 will be installed automatically.
Do you want to install IE 5.5 SP2 too? [ y / n ]
```

```
And do you want to install IE 5.01 SP2? [ y / n ] #這兩個選項都直接按 Enter
```

```
IEs can be installed using one of the following locales:
```

```
EN-US PT-BR DE FR ES IT NL SV JA KO NO
```

```
DA CN TW FI PL HU AR HE CS PT RU EL TR
```

```
Default is . Hit enter to keep it or choose a different one:
```

```
TW #輸入TW ( 必需大寫 )
```

```
By default, I will install everything at /home/ducati/.ies4linux
```

```
I will also install Flash 9 plugin and create Desktop shortcuts.
```

```
Is that ok for you? (To configure advanced options type n) [ y / n ]
```

```
#按下 Enter, IEs 4 Linux 會下載並安裝 IE 所需的檔案。
```

這樣就安裝完成囉！而且可以在桌面上看到 IEs 4 Linux 的圖示



圖 11.1 IE 圖示

啟動 IE



圖 11.2 可以直接看有 Flash 的網站

看 IE only 的網站

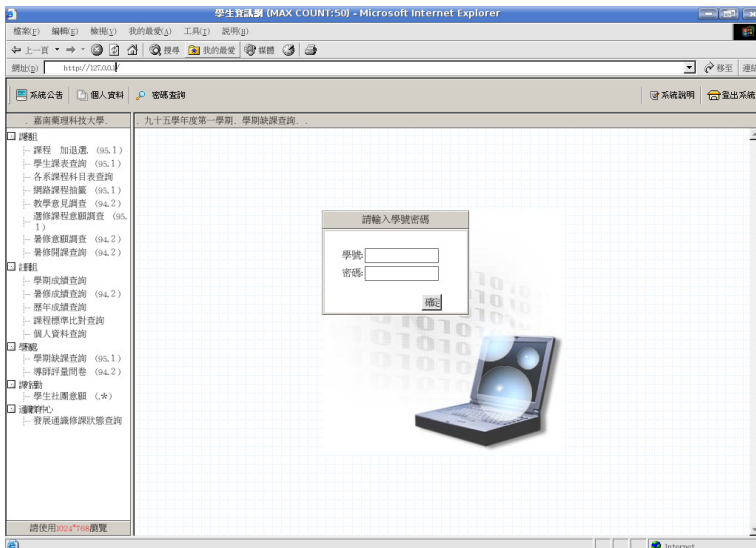


圖 11.3 IE only 的網站

11.3 用 Wine 來裝上你的魔獸世界

你沒看錯，魔獸世界的的確確可以在 Linux 上玩，而且筆者覺得效能也比較好。進入正題，可以先到這個網頁 <http://appdb.winehq.org/appview.php?iAppId=1922>，選擇適合的版本參考安裝，筆者在寫的時候魔獸世界已經更新到 1.12 版了，不過筆者到便利商店買還只能買到 1.11，不過安裝上其實都差不多，基本上可以選擇最新版的網頁參考，因為 Wine 會追著魔獸世界跑，所以我們要選最新版的 Wine，連 patch 也會比較新，筆者寫這個章節時，是參考 <http://appdb.winehq.org/appview.php?iVersionId=5606>，這是 1.12.x 版的裝法，網頁裡有 HOWTO，安裝時也可以參考，寫的很詳細。雖然當時筆者到便利商店買時，魔獸世界的安裝光碟只有出到 1.11 版，不過沒關係！看 1.12.x 的網頁安裝也沒關係，裝好魔獸世界後，讓它自己更新就好了。不過開始前請參考第六章，將你顯示卡的驅動程式裝好。以下之描述以 wine-0.9.22 版為例，至於新的版本，由於更新速度很快，不見得適用書中提到的方式。

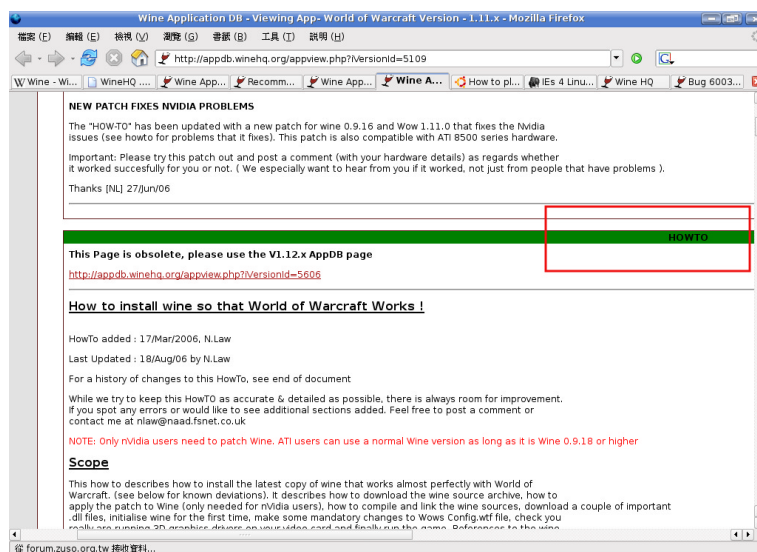


圖 11.4 魔獸世界 HOWTO

11.3.1 編譯你專屬的 Wine

第一步 build-dep

這一步是一定要的哦！因為要確保等一下 Wine 可以編譯成功順利安裝。

```
$ sudo apt-get build-dep wine
```

接著看一下 gcc 是不是有裝上去了，應該會出現像這樣的版本訊息

```
$ gcc --version
ducati@ubuntu:~$ gcc --version
gcc (GCC) 4.1.2 20060928 (prerelease) (Ubuntu 4.1.1-13ubuntu4)
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

第二步 get Wine source

```
$ apt-get source wine
```

在這裡要記得哦！在哪個目錄下 get source, 下載回來的 source code 就在哪。像筆者自己是另外再建個目錄 wine-patch-build.deb, 然後才在裡面 get source, 這樣比較方便管理，抓下來後會自己解壓縮，會多個目錄，如筆者自己抓下來的 source 是 0.9.22 版，所以會有個 wine-0.9.22 的目錄，其他還有三個檔案。註：在這裡要注意，抓下來的 source 不見得是 0.9.22 版，也有可能是更新的。

第三步 打上 patch

你可以將 steam 和魔獸世界的 patch 一起打上哦！steam 的 patch, <http://bugs.winehq.org/attachment.cgi?id=3432&action=view>, 直接下載並存成 steam.diff, 接著是魔獸世界的 patch, <http://stu.phys.ncku.edu.tw/~tux1/wine-wow-0.9.21.diff>, 基本上這個 patch 是給 nvidia 顯示卡晶片的 user 用的，雖然是 0.9.21 的 patch 不過 0.9.22 一樣適用，如果你是 ATI 或其他廠牌可以不用裝。而這兩個 patch 檔，要放到你剛才抓下來的 wine-0.9.22 目錄裡，並在 wine-0.9.22 目錄底下輸入以下指令。

```
ducati@ubuntu:~/program/wine-patch-build.deb/wine-0.9.22$ patch -p1 <
wine-wow-0.9.21.diff
patching file dlls/winex11.drv/opengl.c
```

```
ducati@ubuntu:~/program/wine-patch-build.deb/wine-0.9.22$ patch -p1 <
steam.diff
patching file dlls/mshtml/htmltextcont.c
```

打上 patch 後先改一下 changelog，可以改大一兩個版本，這樣你在更新套件的時候 Wine 才不會被蓋掉，因為被蓋掉很麻煩，我們自己編的 Wine 是有經過 patch 的，假如被蓋掉的話，魔獸世界可能就不能玩了，日後魔獸世界也會更新，或是出資料片也一樣是差不多這樣的安裝方法，讀者也可以參考 11.3 章節的第一個網址，來裝其他版本的魔獸世界。changelog 在 wine-0.9.22/debian 裡面，你可以用 gedit 來編輯。

```
ducati@ubuntu:~/program/wine-patch-build.deb/wine-0.9.22/debian$ gedit
changelog
```

然後將第一行的版本編號，改大一兩個數字，如原本的是 0.9.22 改成 0.9.24

```
wine (0.9.24-0ubuntu1) edgy; urgency=low

 * New upstream release (from upstream)
The usual assortment of MSI improvements.
.... 以下省略
```

第四步 build package

一樣是在 wine-0.9.22 目錄下，這個過程會花蠻多時間的，請耐心等待。

```
$ sudo dpkg-buildpackage
```

編好的 .deb 檔會有四個，會出現在 wine-0.9.22 的上一層目錄裡。

第五步 裝上 Wine

先到 wine-0.9.2.2 的上一層目錄，你可以看到有四個 .deb 檔，而且都是 0.9.24 版的，都把它裝上去吧！

```
$ sudo apt-get install *.deb
```

11.3.2 建立 .wine 資料夾

```
$ winecfg
```

這個指令可以叫出 wine 的設定工具，並且在你的家目錄下建立一個 .wine，資料夾裡面會包含有 “drive_c” 目錄，然後 drive_c 裡面又會有 “Program Files”，“windows” 這兩個目錄，也就是說 “drive_c” 就等同於是 Windows 裡面的 「c:\」。

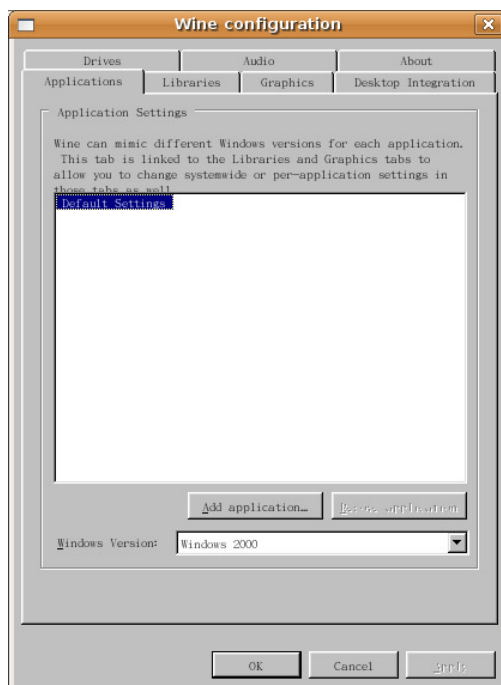


圖 11.5 Wine Configuration

11.3.3 下載安裝魔獸世界所需要的 dll 檔

然後將安裝魔獸世界所需要的 dll 檔，下載到你家目錄下的 .wine/drive_c/windows/system32 這個目錄裡，共有三個。

- mfc42.dll : <http://www.dll-files.com/dllindex/dll-files.shtml?mfc42>

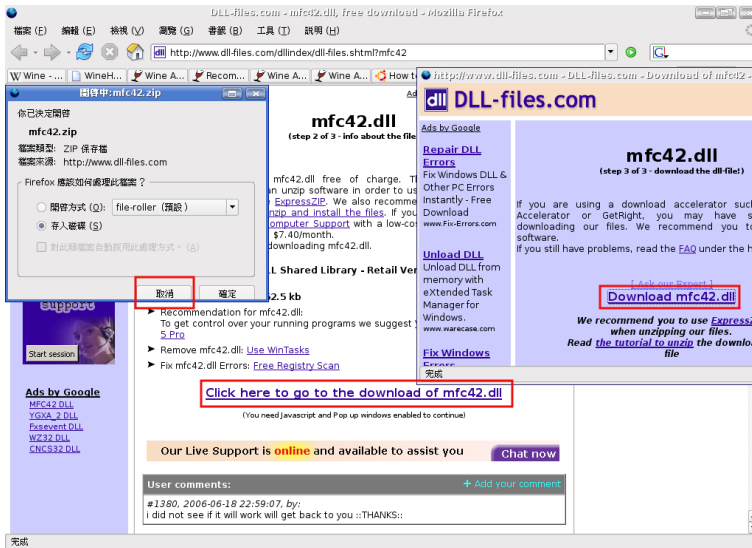


圖 11.6 下載 mfc42.dll

● riched20.dll : <http://www.dll-files.com/dllindex/dll-files.shtml?riched20>

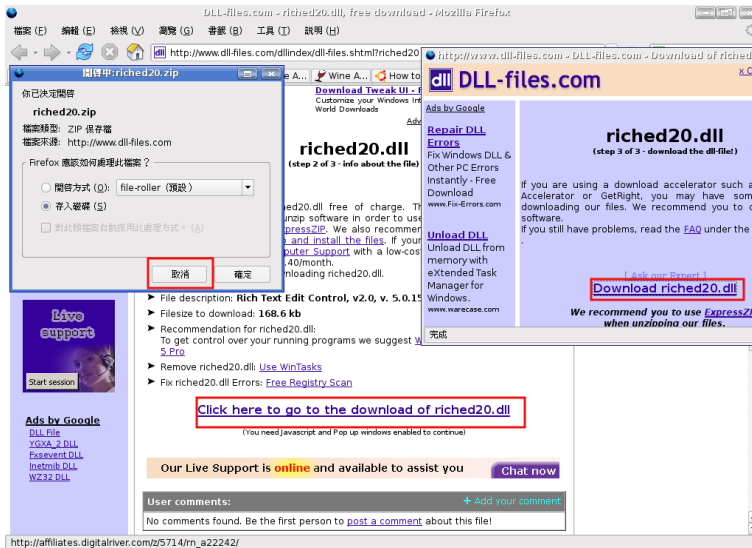


圖 11.7 下載 riched20.dll

- riched32.dll : <http://www.dll-files.com/dllindex/dll-files.shtml?riched32>

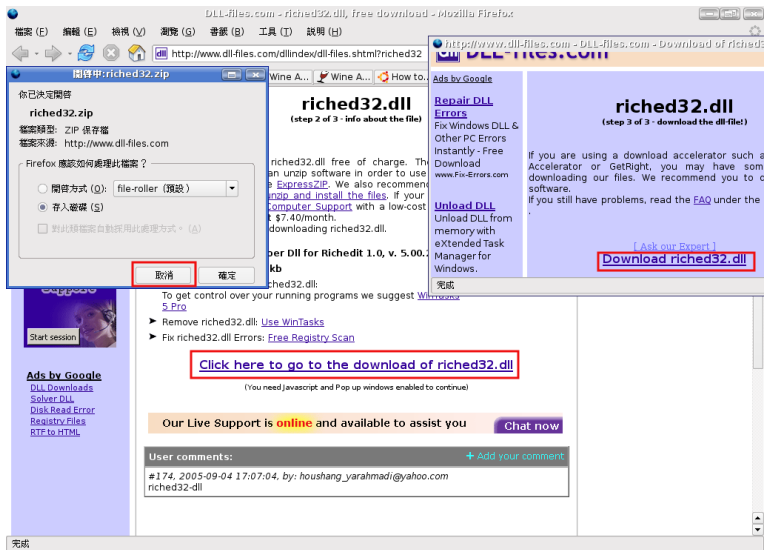


圖 11.8 下載 riched32.dll

11.3.4 開始安裝魔獸世界

光碟片放入後會出現光碟裡的檔案，接著在 Installer.exe 上按右鍵，並選擇「以 Wine Windows Emulator 開啟」。或是用 console 到 /media/cdrom0，還是 /media/cdrom1 裡面輸入 wine Installer.exe。

11.3.5 魔獸世界安裝程式



圖 11.9 開始安裝魔獸世界

按下去後要等一些時間，才會出現安裝畫面，不要以為是當掉了哦。如果真的很久還是開不起來，就重新開機再試一次。接著會出現使用者合約，不過都是亂碼，沒關係如果你還是同意的話就按下去吧。



圖 11.10 使用者合約

接著會問你要安裝在哪個目錄，用預設的就行了直接按好，但是要注意你的硬碟空間夠不夠，安裝魔獸世界保險一點，最少也要 5.5 GB。



圖 11.11 安裝目錄

再來就是漫長的安裝過程，至少也要半小時吧！

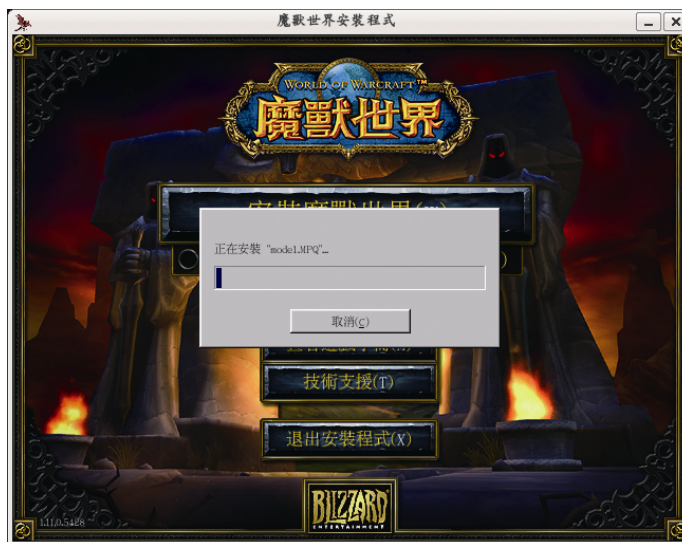


圖 11.12 正在安裝

安裝好會出現警告，說是沒有足夠的空間，如果空間還夠的話，就別理它按好。



圖 11.13 警告安裝空間

如果你先前已經有先確認過硬碟空間大小，那麼直接按下好即可。



圖 11.14 魔獸世界安裝完成

接下來新增一下設定檔。

11.3.6 設定以 OpenGL 開啟遊戲

要再設定一下設定檔，讓魔獸世界用 OpenGL 的模式執行程式。

```
$ gedit ~/.wine/drive_c/Program\ Files/World\ of\ Warcraft/WTF/Config.wtf
```

貼上以下設定

```
SET gxApi "opengl"  
SET ffxDeath "0"
```

11.3.7 音效設定

如果沒有音效的話，再加上下面的設定值。

```
SET SoundOutputSystem "1"  
SET SoundBufferSize "150"
```

或是

```
SET SoundBufferSize "232"
```

看看你的音效卡是不是有被抓到

```
$ lspci |grep Multimedia  
  
0000:00:1f.5 Multimedia audio controller: Intel Corporation 82801DB/DBL/DBM  
(ICH4/ICH4-L/ICH4-M) AC' 97 Audio Controller ( rev 02 )
```

你的音效卡不見得跟筆者我的一樣哦。

11.3.8 3D 加速

接下來就測試看看 3D 加速有沒有啟動

```
$ glxinfo | grep rendering  
direct rendering: No
```

通常可能會沒有，那是因為 .wine 的關係，所以你必需要重新啟動 gdm，這樣 direct rendering 才會啟動變成 Yes，如果這裡是 Yes 的話，你可以直接點桌面上的圖示啟動魔獸世界。

```
$ sudo /etc/init.d/gdm stop
```

這時會只有文字介面，輸入帳號密碼後，在 console 下輸入 startx

```
$ startx
```

然後呢再開啟 console，輸入

```
$ glxinfo | grep rendering
direct rendering: Yes
```

這樣 direct rendering 就會變成 Yes 囉，如果還是 No 的話，那代表你的顯示卡驅動程式可能是沒裝好，或是 3D 加速沒有啟動，可以參考第六章安裝。

11.3.9 啟動魔獸世界

這時候就可以用桌面上的魔獸世界小圖示啟動，或是在 console 下輸入：

```
$ wine ~/.wine/drive_c/Program\ Files/World\ of\ Warcraft/WoW.exe
```

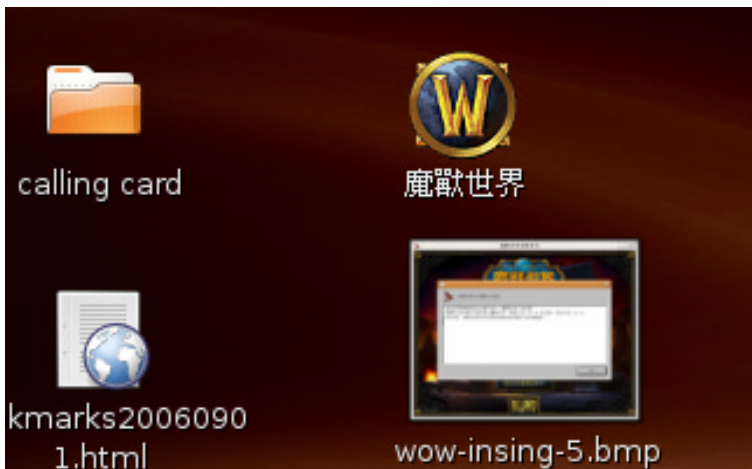


圖 11.15 魔獸世界啟動圖示

欣賞完開場動畫後，並同意最終用戶許可協議和服務條款後，就會出現登入畫面啦。



圖 11.16 魔獸世界登入畫面

下載更新並重新啟動



圖 11.17 下載更新

然後會出現暴雪下載程式，下載更新 patch 檔中



圖 11.18 暴雪下載程式

下載完後開始更新地圖檔與其他檔案中

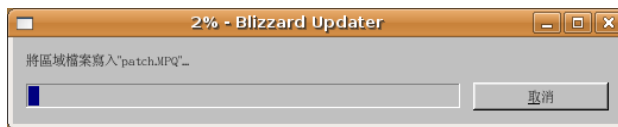


圖 11.19 Blizzard Updater

接著會出現 Gecko Installer, 選擇 Install 安裝



圖 11.20 Gecko Installer

接著按下開始遊戲，不過還要再更新一次才可以玩。



圖 11.21 開始遊戲

然後輸入完你的帳號密碼後，就可以開始在魔獸世界中奔馳啦！yeah!



圖 11.22 遊戲畫面

不過因為 Wine 還是有 bug，所以在遊戲中用 scim、gcin 輸入中文會有問題，所以玩魔獸時筆者建議你可以先裝 oxim 替用。oxim .deb 檔下載網址：<ftp://140.111.128.66/odp/OXIM/Distribute/Ubuntu/>，請下載：oxim_1.1.1-0ubuntu1_i386.deb、oxim-gtk2-immodule_1.1.1-0ubuntu1_i386.deb 這兩個套件回來安裝即可，其他套件可以裝也可以不裝。

✎ 安裝 oxim

```
$ sudo dpkg -i oxim_1.1.1-0ubuntu1_i386.deb  
$ sudo dpkg -i oxim-gtk2-immodule_1.1.1-0ubuntu1_i386.deb
```

✎ 用 im-switch 啟用 oxim

```
$ im-switch -s oxim
```

接著登出登入後，就可以在魔獸世界中輸入中文了



圖 11.23 輸入中文

11.4 用 Wine 安裝 Steam

基本上安裝 Steam 的步驟，都和安裝魔獸世界差不多，在安裝魔獸世界的章節時，有說明怎麼將 Steam 的 patch 給安裝上去。按照完上面的步驟裝好 Wine 及 IE 後（因為安裝 Steam 時，需要用到 Mozilla ActiveX control，而我們在安裝 IE 的時候就會自動裝好了），上面的步驟都做好了以後，便下載 Steam 所需要的字型檔 tahoma.ttf，這個字型是包含在 Microsoft core fonts，你可以從你正版的 XP 裡拿過來用，或是用 google search 一下 tahoma.ttf keyword，印象中第一筆資料裡就找得到，並放到 `~/.wine/drive_c/windows/fonts/` 目錄下，就可以進行安裝囉！

讀者可以參考 <http://www.linux-gamers.net/modules/wiwimod/index.php?page=HOWTO%20Steam>

11.4.1 Steam 主程式下載


你可以到 <http://www.socgame.com.tw/> 這個網站找到，keyword steam。下載回來後的檔案名稱，應該要是 SteamInstall.exe。

Steam 主程式：<http://www.socgame.com.tw/gogogo.php?fid=58&url=1>

安裝 Steam

先到你下載 Steam 主程式的目錄下，輸入

```
$ wine SteamInstall.exe
```

 選擇英文安裝

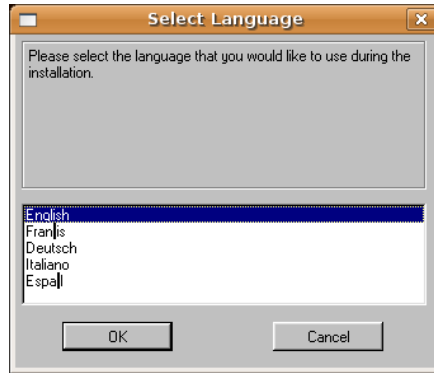


圖 11.24 選擇英文安裝

是否同意授權



圖 11.25 是否同意授權

中間略過，再來選擇安裝目錄，預設的就可以了

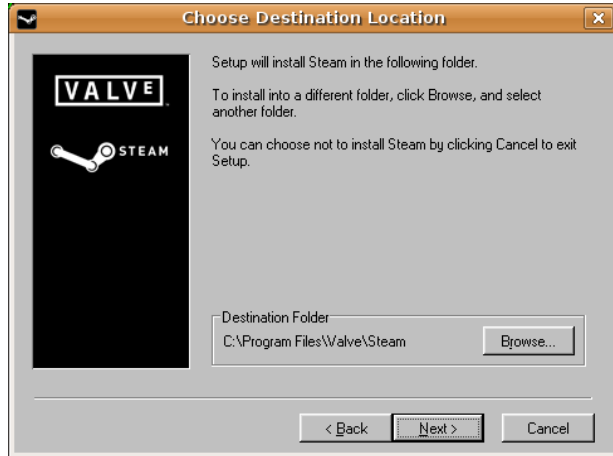


圖 11.26 選擇安裝目錄

一樣選英文



圖 11.27 語言

開始安裝

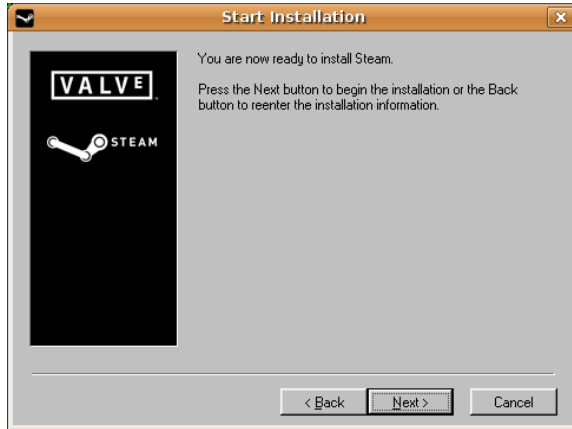


圖 11.28 開始安裝

安裝 Steam 中

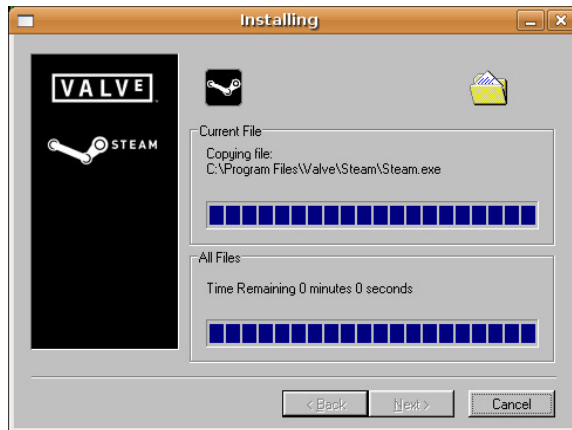


圖 11.29 安裝中

Steam 更新

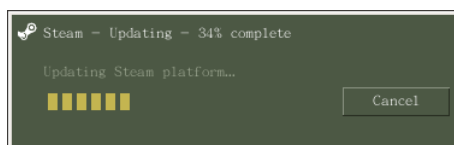


圖 11.30 更新中

接下來就看讀者，是否要新建帳號，或是直接登入了



圖 11.31 新建帳號或直接登入

由於筆者手邊沒有 cs 的 cd key，所以無法提供 CS 的遊戲畫面，僅此致歉。

CHAPTER 12

Linux 遊樂場

誰說 Linux 沒有遊戲？其實 Linux 可以玩的遊戲很多，除了 Linux 自己的遊戲，其實也有很多 windows 遊戲發行 Linux 版本，例如：Enemy Territory。而事實上 Linux 也有模擬器可以玩，不過因為這個章節是 Linux Games，所以在這裡就先不介紹囉！那麼在你開始閱讀這個章節前，建議你可以先參考第六章，將顯示卡的驅動程式裝好，以確保遊戲可以順利執行。

12.1 PPracer，3D 企鵝滑雪

這款 3D 企鵝滑雪，超可愛的小企鵝，是 Linux 上的老遊戲了，也可以用來測試你的 3D 加速。

✎ 安裝方式

```
$ sudo apt-get install planetpenguin-racer
```

✎ 開啟 PPracer

在 console 下輸入

```
$ ppracer
```



圖 12.1 PPracer

12.2 Tuxkart，3D 的小企鵝賽車

這個也是超可愛的老遊戲，有衝刺跳板，還有火箭炮可射哦！

✎ 安裝方式

```
$ sudo apt-get install tuxkart
```

✎ 開啟 Tuxkart

```
$ tuxkart
```



圖 11.2 Tuxkart

12.3 Supertux，超級瑪琳？超級企鵝！

小時候一定玩過超級瑪琳吧？Linux 下有超級企鵝，玩起來一定不陌生，也可以變大和丟火球哦。

✎ 安裝方式

```
$ sudo apt-get install supertux
```

✎ 啟動 Supertux

可以在 應用程式--> 遊戲--> SuperTux，啟動哦！或是在 console 啟動。

```
$ supertux
```



圖 12.3 SuperTux

12.4 Armagetron，類似貪食蛇的 3D 線上遊戲

遊戲方法是車子起跑後，就會沿著你跑出的路線產生圍牆，然後要把你的對手圍住就贏了，不過對手都很強，筆者自己試玩一下，還蠻耐玩的。

✎ 安裝方式

```
$ sudo apt-get install armagetron
```

✎ 啟動 Armagetron

可以在 應用程式--> 遊戲--> Armagetron，啟動。也可以在 console 下輸入


```
$ armagetron
```

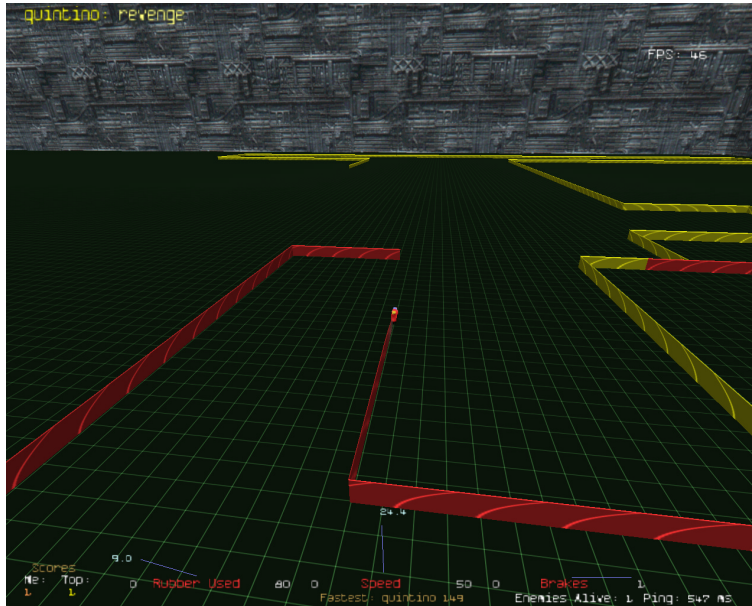


圖 12.4 Armagetron

操作方式是以 z、x 來控制方向。

12.5 bzflag, 3D 線上戰車遊戲

這個蠻有趣的，不過筆者不太會玩，好像是要搶旗子，然後還要把對手幹掉，不太好上手，有興趣的讀者可以試試哦！

🔧 安裝方式

```
$ sudo apt-get install bzflag
```

🔧 啟動 bzflag

可以在 應用程式--> 遊戲--> bzflag，啟動。或是在 console 下輸入

```
$ bzflag
```

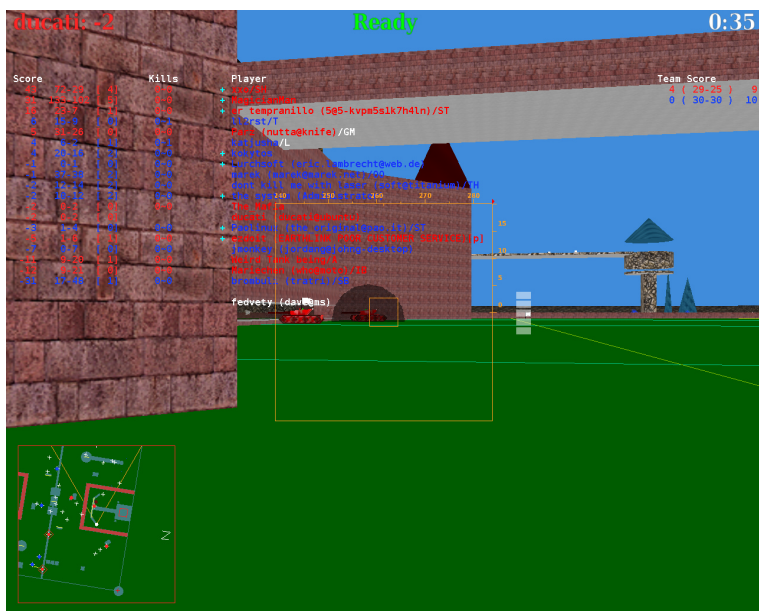


圖 11.5 bzflag

12.6 Tremulous，3D Online FPS

12.6.1 Tremulous 簡介

這是一個使用《雷神之鎚》（QUAKE3）引擎所製作的 GAME，同時也是自由軟體，筆者試玩的結果還不錯，畫面很漂亮玩起來也很順，最重要的是可以上 Internet 玩。遊戲的故事背景筆者不是很清楚，不過玩法跟 cs 蠻像的，一開始要選擇要當 human 或是 alien，如果你選的是 human，那還會有兩個選項，第一個是傭兵可以一直殺 alien，然後買一堆有的沒的怪武器，也可以買飛行器在空中飛，第二個是工兵可以在自家蓋機槍，醫療站等等。alien 也是一樣可以選兩個，而 alien 應該是殺越多的人，就會變得更強更大隻，也有分蟲種，也可以選擇工蟲，蓋一些像大便一樣的東西。

12.6.2 安裝 Tremulous

方法一

如果你用的 Ubuntu 不是 6.10 版本，可以到 Tremulous 的官方網站，下載最新的安裝檔，網址 <http://tremulous.net/index.php?section=files>，選擇 .run 的 Linux 執行檔。

下載回來後改一下檔案權限，然後安裝

```
$ chmod +x  
tremulous-1.1.0-installer.x86.run ; ./tremulous-1.1.0-installer.x86.run
```

授權

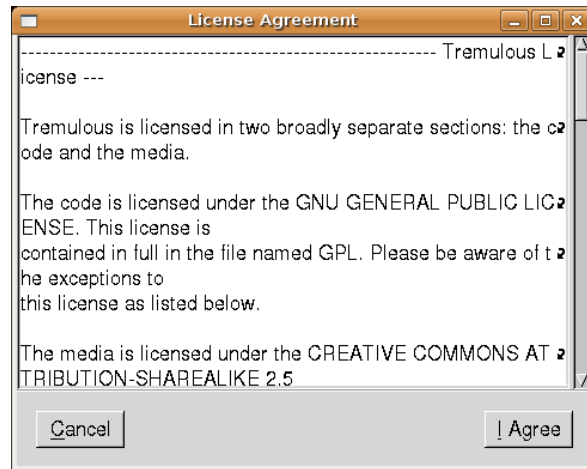


圖 12.6 是否同意授權

同意的話就可以繼續安裝，接著會問你要安裝在哪個目錄

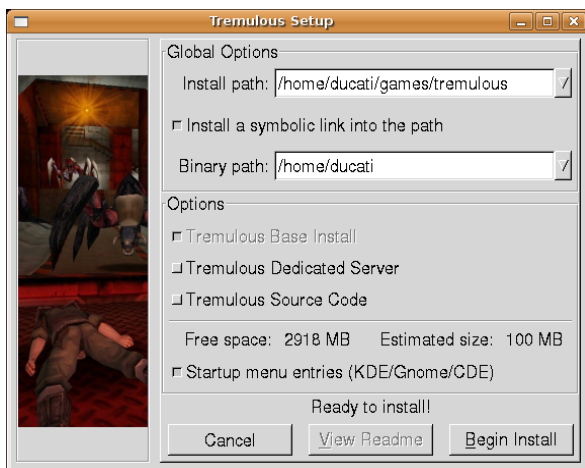


圖 12.7 安裝目錄

這裡的 binary path 用預設的就可以了

✎ 安裝進度



圖 12.8 安裝進度

開始遊戲

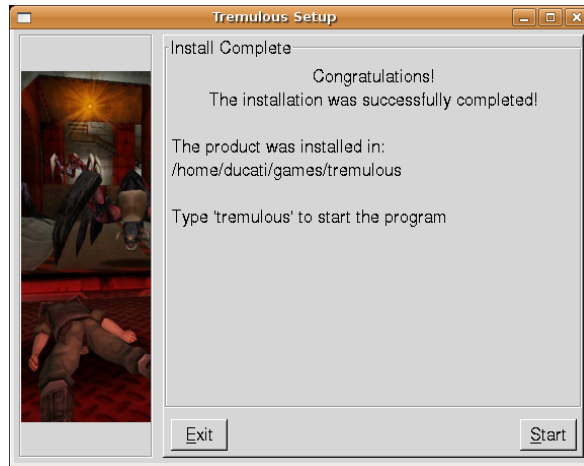


圖 12.9 開始遊戲

遊戲畫面



圖 12.10 遊戲畫面

開啟 Tremulous

先到你安裝 Tremulous 的目錄下，然後輸入

```
$ ./tremulous.x86
```

方法二

在 ubuntu 6.10 版本可直接 apt-get

```
$ sudo apt-get install tremulous
```

可以在 **應用程式--> 遊戲** > Tremulous，開啟 Tremulous。不過要記得開始遊戲後，要調解析度，不然遊戲畫面會很小，筆者自己是調成 1280x1024。

12.7 Enemy Territory，網路 FPS 遊戲

相關網頁

- <http://www.mrbass.org/enemyterritory/>
- <http://www.truecombat.us/tce.php?page=downloads>

還有其他的 Mods 可以下載哦！像是 TC:E 也不錯玩，和 cs 一樣是拿現代武器。

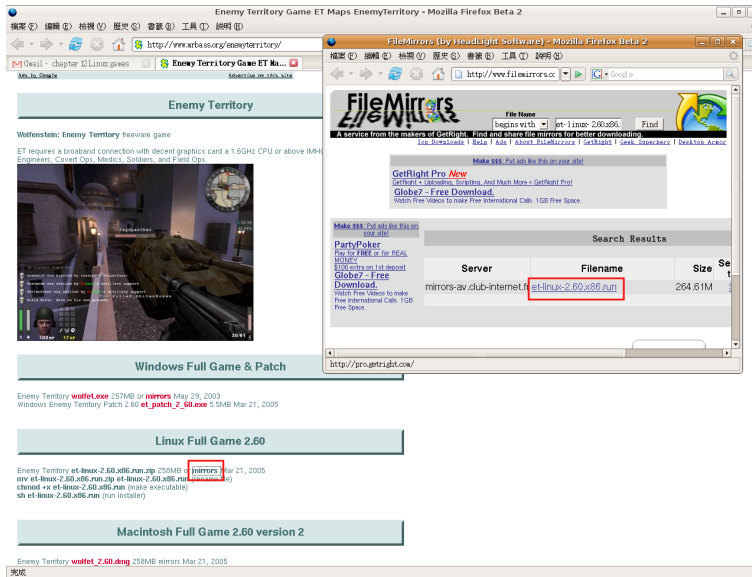


圖 12.11 Enemy Territory 下載網頁

For Linux 的下載網址

<http://mirrors-av.club-internet.fr/pub/games/nofrag/returntocastlewolfenstein/n/et-linux-2.60.x86.run>

到你下載安裝檔的目錄下，接著修改檔案權限，讓檔案可以執行

```
$ chmod +x et-linux-2.60.x86.run
```

開始安裝

```
$ sh et-linux-2.60.x86.run
```

接著需要輸入 root 密碼，請讀者先設好 root 密碼，可用 `sudo passwd root` 設定。

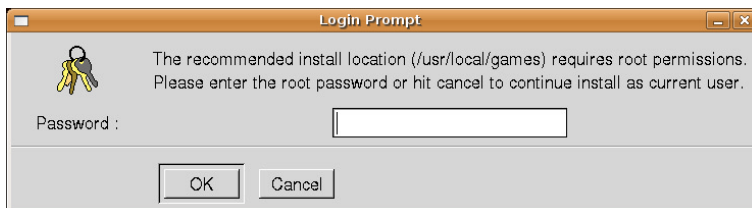


圖 12.12 輸入 root 密碼

接下來的選項都按 Enter 就可以了，然後到選擇安裝目錄這個選項，預設是安裝到 /usr/local/games/ 底下，接著就可以安裝到底啦！

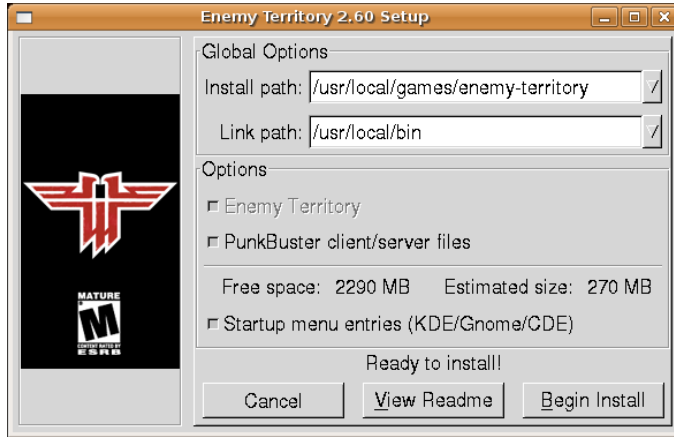



圖 12.13 選擇安裝目錄

遊戲畫面



圖 12.14 Enemy-Territory

 開啟遊戲

如果你是用預設的安裝，請用 console 到 `/usr/local/games/enemy-territory/` 目錄下，接著輸入

```
$ et &
```

假如遊戲進行中沒有聲音的話，請到 console 下切換成 root

```
$ su
```

輸入 root 密碼，接著輸入：

```
# echo "et.x86 0 0 direct" > /proc/asound/card0/pcm0p/oss
```

如果還是沒聲音，那麼重開機後再執行一次遊戲即可。

CHAPTER 13

深入瞭解 Ubuntu 系統

要深入瞭解 Ubuntu Linux 系統的架構，勢必得需要先知道系統在開機的時候跑過哪一些流程，這樣在系統出現錯誤時，我們可以依照流程圖判別系統是在哪裡出現錯誤，並找出除錯方案。所以這一章會先簡介開機的流程，接下來把這些流程的每一個細部動作拿出來講解討論。

13.1 開機流程

基本上所有的 Linux 發行版開機流程都差不多，有些差異的地方可能是設定檔放置位置不同，所管理的工具程式不一樣。不過大體上觀念是一樣的，所以這裡弄懂的話，以後要學習別種 Linux 發行版也是很容易的。

當電腦電源開啟時，系統會把控制權全部交給 BIOS，而 BIOS 會開始對硬體作檢查並且初始化。接下來 BIOS 會去找硬碟的 MBR (Master Boot Record) 磁區的開機程式，並且去執行它！由於 MBR 的大小只有 512 bytes，在今天的複雜應用其實是無法在小小的 MBR 放上完整的 bootloader 開機程式，所以在 Linux 上最常用的 bootloader GRUB 就拆成 stage 1 和 stage 2，其中的 stage 1 就放在 MBR 上。當 stage 1 上的 bootloader 從 BIOS 取得控制權後，它會先驅動硬碟，接下來從硬碟內找到 stage 2 的程式，並執行 stage 2 的 bootloader。

Stage 2 的 bootloader 執行後，會先找 `/boot/grub/menu.lst` 去取得可用的核心和系統。接下來顯示並讓我們選擇使用的核心或者改成從 Windows XP 開機的選單。若讀者選擇啟動 Linux 的話，它會依照 `menu.lst` 內的核心檔案位置去載入該核心映像檔，接下來該核心會開始初始化硬體並依照 `fstab` 把該掛載的硬碟或者光碟機等檔案設備掛載進來。

為了減少核心大小，有一些硬體驅動程式會編成模組而不會全部塞到核心裡面。此時系統還有一些額外的硬體還沒有驅動，所以核心就會把這些硬體所需要的模組裝起來。最後核心會執行 `/sbin/init` 這個程式，這個程式會讀取 `/etc/inittab` 裡面的設定，並且進入 `runlevel`，跑完這些 `script` 後，系統已經正式的開機完成了。

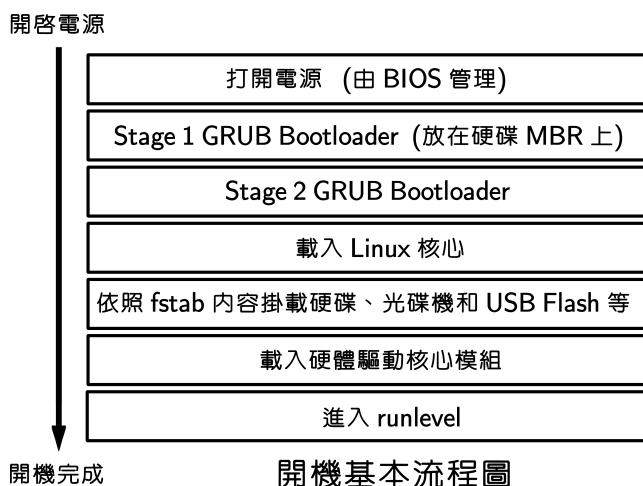


圖 13.1

13.2 `fstab` 開機磁碟掛載與 UID 新架構

`fstab` 是一個開機的時候核心應該要掛載哪些硬碟分割區或光碟等等的設定檔。例如我們開機後想要把 `hda1` 掛載到 `/`，`hda2` 掛載到 `/home` 就是把設定寫到這個檔案。傳統上我們會在 `/etc/fstab` 裡面直接指定 `hdaX` 掛載到某個目錄底下，但是在 Ubuntu Edgy 引入了 UID 的新架構，這可以讓某些軟體 Raid 等應用不會因為某台硬碟出錯而導致硬碟編號改變，造成 Raid 系統錯誤！

13.2.1 為何引入 UUID 與新的管理方式

要談論 UUID 引進的新架構，我們勢必得先溫故知新。在以前設定 fstab 的時候，大致上如以下範例的內容一樣。在該設定檔中，第一個欄位代表了某個磁碟裝置，第二個欄位代表了該設備將要被掛載到的位置，第三個欄位代表該分割區或裝置所使用的檔案格式。最後在加上一些掛載參數和是否開機時要 fsck 檢驗和備份時要不要 dump 到。

 傳統的 /etc/fstab 設定檔：

#	<檔案系統>	<掛載點>	<格式>	<參數>	<dump>	<pass>
	/dev/hda4	/	reiserfs	defaults	0	1
	/dev/hda3	/boot	ext3	defaults	0	2
	/dev/hda6	/home	reiserfs	defaults	0	2
	/dev/hda5	none	swap	sw	0	0
	/dev/hdc	/media/cdrom0	udf,iso9660	user,noauto	0	0

檔案系統的設備是 SCSI 或者 SATA 的時候，設備代號是 sdX 系列的，它會依照順序排列下去。當哪天有個設備出現問題時，使用者或許會先把它拔掉，這樣設備代號順序不會有空缺，而是由後面的遞補。這時候就有大麻煩啦，此時 fstab 裡面的設定就會和實際硬體的設備位置不一樣，導致系統無法開機。所以 IDE 硬碟不會遇到這種問題，畢竟每一個 IDE 插槽都對應到個別的代號。

筆者在以前實際有遇到和這種問題相關的錯誤。筆者原來的伺服器使用 SCSI 硬碟，之後遇到空間不足，但是因為實驗室沒有足夠的經費，所以改成添加 SATA 硬碟。原本的 SCSI 硬碟代號是 sda，但是在此 Server 上不知道為什麼 SATA 的優先權比較高，在裝上了 SATA 硬碟後，原本的 SCSI 設備變成 sdb，而新的 SATA 設備代號為 sda。這樣系統上 fstab 甚至是 GRUB 裡面的裝置位址都會出現錯誤，在開機的時候就會找不到核心，系統出現錯誤！這種問題也有可能發生在軟體 Raid 系統，當一顆硬碟故障後，若把故障的硬碟拆下來後沒有換新的，可能後面的硬碟會遞補上來，重新開機後系統會以為後面那一顆是新的陣列用硬碟，所以系統會去修復 Raid 系統，此時會導致被遞補上來的硬碟資料損毀。


為了解決這種問題，Ubuntu Edgy 開始引入了 UUID 的功能，它會在每個分割區和硬碟產生一個唯一的識別碼，所以在 fstab 裡面不再使用 /dev/hdaX 來指定

裝置，而是使用該裝置的 UUID 當成裝置代號，當然使用傳統的方式也是可以被允許的，而預設安裝好後是採用 UUID 的方式。

 在 edgy 引入 UUID 的 /etc/fstab：

# <檔案系統>	<掛載點>	<格式>	<參數>	<dump>	<pass>
# 取代原本的 hda4					
UUID=956cab3e-a68b-45c0-9350-7f0da4810e02	/	reiserfs	defaults	0	1
# 取代原本的 hda3					
UUID=633055b1-7736-42ad-a73d-8e1f59b6737a	/boot	ext3	defaults	0	2
# 取代原本的 hda6					
UUID=9bd96654-3ac0-4027-9724-d6219fa17f82	/home	reiserfs	defaults	0	2
# 取代原本的 hda5					
UUID=af7a979a-a383-441c-a865-9efab61b8cce	none	swap	sw	0	0

所以在 Ubuntu edgy 以後，使用者可以採用新的設定方式，或者傳統的明確指定方式。當然，在 SCSI 介面的硬碟中，筆者建議還是採用 UUID 的設定方法。

 查詢硬碟的 UUID：

```
dbtsai@ubuntu:~$ ls /dev/disk/by-uuid/ -lh
(略) 633055b1-7736-42ad-a73d-8e1f59b6737a -> ../../hda3
(略) 956cab3e-a68b-45c0-9350-7f0da4810e02 -> ../../hda4
(略) 9bd96654-3ac0-4027-9724-d6219fa17f82 -> ../../hda6
(略) af7a979a-a383-441c-a865-9efab61b8cce -> ../../hda5
```

讀者可以看到其實這些 UUID 會 soft link 到該硬碟，所以可以透過以上指令來查詢。若使用者的硬碟順序改變了，它也會在開機的時候自動把這裡的對照表更新，因此不會造成磁碟編號錯亂。

13.2.2 使用 NTFS-3G 讀寫 NTFS 分割區


預設 Ubuntu 安裝後，若有抓到 MS Windows 的 NTFS 分割區，會使用 kernel 裡面內建的 ntfs module 來驅動，但目前這個模組還無法對 NTFS 分割區作寫入的動作，導致使用上非常不便。

最近有一套 user space 的 NTFS 分割區的驅動程式被開發出來，實際上它的開發者和 kernel modules 那個驅動程式是同一批人。在 user space 上的驅動程


式比較好除錯，也比較不為因為稍微的錯誤就導致系統掛了，因此他們才先開發出 user space 的驅動程式。雖然 kernel module 會有比較好的效能，但是開發難度比較高，容易因為一點程式碼的錯誤導致核心當機，所以他們先針對 NTFS-3G 作開發，之後會移植到 kernel module 裡面。

在筆者的測試中，NTFS-3G 已經有不錯的效能，且在日常使用上非常穩定，尤其是在讀者有使用雙系統的狀況下，可以很容易的去讀寫 MS Windows 的分割區。

若您的核心是自己編譯的，會因為缺少了 fuse 模組，而無法使用以下方式安裝，此時就需要從原始碼安裝。

 從套件庫安裝 NTFS-3G（若您的核心是使用 Ubuntu 內建的，請使用這個方法）：

```
dbtsai@ubuntu:~$ sudo apt-get install ntfs-3g
```


 從原始法安裝 NTFS-3G（若您的核心是自己編譯的，請使用這個方法）：

```
dbtsai@ubuntu:~$ mkdir fuse && cd fuse
dbtsai@ubuntu:~/fuse$ sudo apt-get build-dep fuse-source
dbtsai@ubuntu:~/fuse$ apt-get source fuse-source # 下載原始碼
dbtsai@ubuntu:~/fuse$ cd fuse-2.5.3 # 這個目錄可能會隨版本而變
dbtsai@ubuntu:~/fuse-2.5.3$ sudo dpkg-buildpackage # 開始編譯，大概幾分鐘
dbtsai@ubuntu:~/fuse-2.5.3$ cd ..
dbtsai@ubuntu:~/fuse-2.5.3$ sudo dpkg -i *.deb # 把剛剛編譯好的套件裝起來吧
dbtsai@ubuntu:~/fuse-2.5.3$ sudo apt-get install ntfs-3g # 安裝NTFS-3G
```

接下來修改 /etc/fstab，讓系統使用 ntfs-3g 來掛載 NTFS 分割區。記得要先備份一下 fstab 以免有萬一唷！

 修改 /etc/fstab：

```
# 把所有格式是 ntfs 的修改成爲 ntfs-3g，並且稍微修改一下後面的參數。
# 由於 ubuntu 是使用 zh_TW.UTF-8 的 locale，所以記得加上去，不然中文會出不來！
# UUID=略 /media/hdb1 ntfs defaults,nls=utf8,umask=007,gid=46 0 1 # 如把以上格
式的設定值，修改成下面的，就可以對NTFS 寫入了唷！
UUID=略 /media/hdb1 ntfs-3g defaults,locale=zh_TW.UTF-8,umask=007,gid=46 0 1
```

 接下來重新掛載硬碟：

```
dbtsai@ubuntu:~ $ sudo umount -a && sudo mount -a
```

若重新掛載失敗的話，可能是因為該 NTFS 磁碟忙碌中，若真的找不到辦法，可以重新開機，應該就可以順利在 NTFS 寫入啦！

13.3 Runlevel 原理和設定

傳統上 Linux 採用 SysV 的管理模式，把開機要啟動的服務建立軟連結到 `/etc/rcX.d` 時，開機後就會依照 `inittab` 要求去執行不同的 runlevel。例如 `inittab` 若定義開機後執行 runlevel 5，那麼系統開機後會先去跑 `/etc/rcS.d` 內的指令，接下來去執行 `/etc/rc5.d` 內的指令。因此在 Linux 中若要開機後自動啟用某個服務，我們只需要把該服務的執行檔建立一個軟連結到 `/etc/rcX.d` 裡面，而 X 就視您希望在哪些 runlevel 啟用，例如只希望在 runlevel 5 啟用，那就只需要建立一個 soft link 到 `rc5.d` 即可。詳細內容我們後面再用範例來說明。

而這種 SysV 的方式已經很古老了，它只能依照那些 runlevel 內的 link 依照順序一個接一個去執行。但在今日，有些服務是需要某些其他服務先啟用後才能就啟動，因此我們需要很仔細的排定 runlevel 內的執行順序。也有些服務會和硬體有關，但現在即插即用也很流行，所以我們無法預期開機的時候那些硬體就全部就緒，因此無法在 runlevel 就驅動相關服務。典型的例子就是 `fstab`，在 Linux 中會依照 `fstab` 的內容去掛載硬碟等設備，但開機的時候未必該設備已經裝上去。

為了解決這種問題，Ubuntu 從 edgy 拋棄了 `init` 這種啟用機制，改用了 `upstart` 來啟用或停止這些服務。但是讀者不用擔心，因為世界上百分之九十九的書都是教 `init`，百分之九十九的人也只會使用 `init`，所以 Ubuntu 並沒有直接丟棄 `init` 的設定方式，而是透過 `upstart` 來模擬類似 `init` 的行為。簡單的說，在 `upstart` 裡面引入事件驅動的概念，例如我們裝上一個新的硬碟，就會驅動掛載，而不是在開機的時候才會自動掛載。因此各種其他的服務也能用事件驅動的方式來啟用，如 A 服務要先啟用 B 服務，若 B 服務啟用後，會產生是事件，因此 A 服務也跟著驅動了。當然這些事件驅動的方式在 edgy 還用的不多，相信未來會慢慢改成採用這種方式，也可以增加開機速度，並讓管理更加容易。

那麼在 upstart 是如何模擬 init 的效果呢？主要 upstart 的設定檔放在 /etc/event.d 裡面，我們看一個很簡單的例子。

 /etc/event.d/rc5 :

```
# rc5 - runlevel 5 compatibility
#
# This task runs the old sysv-rc runlevel 5 (user defined) scripts. It
# is usually started by the telinit compatibility wrapper.
start on runlevel-5      # 定義 script 在 runlevel 5 時執行

stop on shutdown        # 定義 script 在關機時不執行
stop on runlevel-2     # 定義 script 在 runlevel 2 時不執行
stop on runlevel-3
stop on runlevel-4

script
    set $(runlevel --set 5 || true)
    if [ "$1" != "unknown" ]; then
        PREVLEVEL=$1
        RUNLEVEL=$2
        export PREVLEVEL RUNLEVEL
    fi
    exec /etc/init.d/rc 5
end script
```

在這個範例中，是使用 upstart 的事件驅動來模擬 runlevel 5。讀者可以看到在範例檔中定義了若要跑 runlevel 5，也就是進到 runlevel 5 這個事件，就開始去跑 script 的地方，而那裡面的指令就會自動去跑 /etc/rc5.d 裡面的內容。

接下來會對傳統的 init 系統做詳細的介紹，目前 Ubuntu 整個運作的方式還是採用 init 的相容模式，而 upstart 雖然取代了 init，但是幾乎 Ubuntu 都運作在 init 相容模式，所以傳統的 init 還是非常重要的。而 upstart 的新功能礙於本書的篇幅不夠，加上目前事件驅動還用的很少，所以就不多介紹了。這個我們希望在下次改版時能夠詳細的介紹（其實因為是 upstart 剛引入，有部份原因也是因為截稿時間的問題啦！讀者別打我啊！）

13.3.1 init.d 內的各種伺服器控制檔

如同前面提到了，開機時系統會依照 inittab 去跑不同的 runlevel，而在那些 runlevel 裡面都只是軟連結到 /etc/init.d 裡面。這樣有個好處，我們可以在不同的 runlevel 共用相同的伺服器控制檔，所以整個架構可以很模組化，並且很容易去修改。

首先我們去看一下 /etc/init.d 裡面所有的檔案吧！

```
dbtsai@ubuntu:~ $ ls /etc/init.d/
apache2          killprocs        rc
appport          klogd            rc.local
atd              linux-restricted-modules-common rcS
bittorrent       makedev         README
bluetooth        mdadm            reboot
bluetooth.dpkg-old mdadm-raid      rmnologin
checkroot.sh     mountnfs-bootclean.sh ssh
console-setup    mysql            vsftpd
hdparm           portmap         waitnfs.sh
x11-common
... (略) ...
```

讀者會發現有些和伺服器相關的控制檔，也有一些系統服務，我們可以透過這一些檔案來控制該服務是否要執行。可以在控制檔後面加上參數 start 來啟用服務，或 stop 來停止服務或者 restart 來重新啟動該服務。也可用 reload 來只載入新的設定檔，或者用 force-reload 強迫載入新的設定檔。當然每個服務所支援的參數不一定一樣，讀者可以直接執行該控制檔還得知此控制檔有支援哪些參數。

控制檔操作範例：

```
dbtsai@ubuntu:~ $ sudo /etc/init.d/apache2 start # 啟用 Apache
dbtsai@ubuntu:~ $ sudo /etc/init.d/apache2 stop # 關閉 Apache
dbtsai@ubuntu:~ $ sudo /etc/init.d/apache2 restart # 重新啟動 Apache
dbtsai@ubuntu:~ $ /etc/init.d/apache2 # 得知此控制檔所有可用參數
Usage: /etc/init.d/apache2 start|stop|restart|reload|force-reload
```

在 Ubuntu 裡面還有另外一個方式來啟用或者關閉在/etc/init.d 裡面的服務，這個方式筆者比較少使用，因為沒辦法用 tab 鍵補齊服務名稱，所以打起來反而比較慢。但是這個方式和直接操作控制檔是完全等效的唷！

使用 invoke-rc.d 範例：

```
# 它可以加的參數和前面使用控制檔的方式相同唷！
dbtsai@ubuntu:~ $ sudo invoke-rc.d apache2 start # 啟用 Apache
```

13.3.2 inittab 預設 runlevel 層級

在開機後，系統會先讀取 /etc/inittab 這個設定檔，決定系統等一下進去哪一個 runlevel。基本上 runlevel 可以想像成系統提供很多種不同的工作模式，我們只要切換 runlevel 就可以跑不同的 runlevel。具體而言是我們可以在 runlevel 5 啟用圖形介面，runlevel 3 不啟用圖形介面，那麼我們可以分別在 /etc/rc5.d 裡面設定啟用圖形介面，而在 /etc/rc3.d 設定不啟用圖形介面，之後我們就可以隨心情修改 inittab 決定下一次開機是使用哪一個 runlevel 啦！

runlevel	目錄	簡介
0	/etc/rc0.d	關機
1	/etc/rc1.d	單人模式
2	/etc/rc2.d	多使用者模式
3	/etc/rc3.d	多使用者模式
4	/etc/rc4.d	多使用者模式
5	/etc/rc5.d	多使用者模式
6	/etc/rc6.d	重新開機


修改開機時預設的 runlevel 很簡單，讀者先如以下範例，編輯 /etc/inittab 吧！

 **/etc/inittab 內容：**

```
# /etc/inittab: init(8) configuration.
# 開機後會進入的 runlevel
id:2:initdefault:

... (略) ...
```

如以上內容，在 Ubuntu 預設的 runlevel 是 2，當然您可以改成別的 runlevel！但是若您改成 6 或者 0 會怎麼樣呢？這時候就慘啦，系統會在開機後跑 runlevel 0 然後又自動關機，或者重新開機！所以不要改錯唷！其實 runlevel 0 和 6 是提供系統關機和開機時使用的。在 Linux 中提供 init 這個程式來改變目前系統的 runlevel。

 **改變系統的 runlevel：**

```
dbtsai@ubuntu:~ $ sudo init 0          # 進入 runlevel 0，等效於關機
dbtsai@ubuntu:~ $ sudo init 6          # 進入 runlevel 6，等效於重新開機
```

在 RedHat 系列等其他 Linux 通常 runlevel 3 是文字模式，runlevel 5 是圖形模式，但是在 Ubuntu 預設 runlevel 2 到 runlevel 5 是一樣的！也就是把 inittab 改成 runlevel 3 並不會進入文字模式！這時候我們就要看一下實際上 runlevel 的運作方式了。

 **觀看 runlevel 2 的運作方式：**

```
dbtsai@ubuntu:~ $ ls /etc/rc2.d/ -l
(略)    S10sysklogd -> ../init.d/sysklogd
(略)    S11klogd -> ../init.d/klogd
(略)    S13gdm -> ../init.d/gdm
(略)    S20vsftpd -> ../init.d/vsftpd
(略)    S91apache2 -> ../init.d/apache2
... (以下略) ...
```

在以上範例中，為了減少頁面，已經拿掉一些了，實際上讀者應該會看到幾十個軟連結到 /etc/init.d 的控制檔。這些軟連結的檔名通常是 S 或 K 開頭的，並且後面有一個數字。若是 S 開頭的話，就會在開機後進入 runlevel 2 時執行，順序依照後面的數字由小到大啟動。同樣的若是 K 開頭的話，會依照數字由小到大關

閉。通常在 runlevel 2 到 5 不會看到 K 開頭的檔案，這些關閉服務的 runlevel 通常只在 runlevel 0 和 6 中出現。在 runlevel 0 中，讀者會發現同時有 K 開頭和 S 開頭的檔案，此時就會先把所有的 K 開頭檔案跑完，再跑 S 開頭的檔案。

若以上面這個例子來說，進入 runlevel 2 後，就會依序啟動 `syslogd`、`klogd`、`gdm`、`vsftpd` 和 `apache2`。讀者或許奇怪，這些 runlevel 內的檔案不是直接軟連結到 `init.d` 裡面嘛？這些控制 script 不是需要加參數才能運作？那麼直接去跑 runlevel 裡面這些檔案，應該是無法給與參數，或者指定要關閉或者啟用，那到底是如何運作的呢？其實在跑 runlevel 的時候，是透過 `/etc/init.d/rc` 這個程式去找 runlevel 內到底有哪些檔案，它看到 `S13gdm` 就會自動翻譯成 `/etc/init.d/gdm start`，所以在 runlevel 裡面，若檔案名稱命名錯誤的話，就會自動被忽略。我們稍微歸納一下 runlevel 的運作規則吧！

1. 進入 runlevel 後，先執行關閉的指令 (K 開頭)，接下來才跑啟動服務的指令 (S 開頭)。
2. 不論是 K 開頭或者 S 開頭都是依照大小，從小的開始執行，並且數字可以重覆。

所以若我們想要在 runlevel 3 不要進入視窗介面，那我們要怎樣做呢？很簡單，我們先把 runlevel 3 進入視窗介面的啟動砍了，就達成啦！

在 runlevel 3 時進入文字介面：

```
# 先砍了這個 soft link，數字可能會在不同版本的 Ubuntu 改變！ GDM 基本上是
# 提供 X Window 的登入介面，所以在開機時不啟動這個的話，就沒有視窗介面啦！
dbtsai@ubuntu:~ $ sudo rm /etc/rc3.d/S13gdm
dbtsai@ubuntu:~ $ vi /etc/inittab
(略)
id:2:initdefault:      # 把這裡的 2 改成 3，下次開機就直接進入 runlevel 3
(略)
# 若在文字模式想要在進入視窗，可以用以下指令
dbtsai@ubuntu:~ $ sudo /etc/init.d/gdm start
```

透過以上的範例讀者可以修改讓系統開機後預設進入 runlevel 3，並且不會啟動視窗介面。這種技巧對於伺服器而言特別有用，因為伺服器幾乎是很少用到視窗介面，把它關了可以節省不少資源，當讀者需要的時候，再用 `/etc/init.d/gdm start` 把它打開。或者讀者想要改回成開機時自動啟用的話，也只要把預設使用的

runlevel 改回成 2。接下來我要在下一節討論如何使用一工具程式來管理 runlevel 目錄內的檔案。

13.3.3 管理某服務是否開機後啟用

在 Ubuntu 裡面修改 runlevel 有以下幾種方式。

- 第一種方式是手工刪除或建立軟連結。
- 第二種方式是使用內建的 `update-rc.d` 在文字模式底下增加或刪除 runlevel。
- 第三種方式是使用 `sysvconfig` 來使用光棒模式設定 runlevel，這程式需要額外安裝。
- 最後一種方式是使用 Ubuntu 內建的 `services-admin` 這個程式在視窗模式下設定。


以上各種方式都可以達到相同的效果，要選擇哪一種方式來設定 runlevel 就看使用者的習慣而定。第一種使用手工的方式是適用於所有的 GNU/Linux 作業系統，也是筆者最常用的方式之一。最後一種方式對於初學者而言可以很簡單的設定好 runlevel，值得讀者嘗試看看。以下將會對這四種方式都稍微作一個介紹，讀者選擇您最上手的方式即可。

這裡還要提醒讀者的是，修改 runlevel 的意思是修改開機的時候預設啟動的服務或程式，因此在讀者修改完 runlevel 後，不會立即生效，需要重新開機後才會生效。因此假設讀者把 `/etc/rc3.d/S91apache2` 砍了，或者用其他工具把 Apache2 從 runlevel 移除，只有在下次開機時才不會啟用 Apache2，而不是把 Apache2 從 runlevel 移除後順便就會讓該服務立即停止！所以若要立即停止的話，就需要額外執行 `/etc/init.d/apache2 stop` 喔！在增加新的 runlevel 也一樣，該服務不會立即啟動，只會在下次開機時預設啟動，因此還是得要手動用控制 script 把它啟用唷！

第二點要提醒讀者的是，預設 Ubuntu 從 runlevel 2 到 5 都是一樣的內容，而預設在 `inittab` 是跑 runlevel 2，建議讀者保持 runlevel 2 不要更動，修改 runlevel

3 等，此時要回復系統原來的狀態就只需要把 runlevel 改回來成 2 即可！如上一節的範例，讓 runlevel 2 有視窗介面，但是 runlevel 3 進入文字介面。


手工建立或移除 runlevel 程序

 觀看目前有哪些服務可以加到 runlevel 裡面：

```
dbtsai@ubuntu:~ $ ls /etc/init.d/
alsa-utils      festival          mountnfs-bootclean.sh  README
anacron         gdm              reboot              apache2            glibc.sh
mysql          hdparm          networking
```

(以下略)

從以上範例中看到的控制 script 可以加到 runlevel 裡面讓它自動啟用唷！

 觀看 runlevel 2 的運作方式（其他的 runlevel 也是使用相同方式）：

```
dbtsai@ubuntu:~ $ ls /etc/rc2.d/ -l
(略) S10sysklogd -> ../init.d/sysklogd
(略) S11klogd -> ../init.d/klogd
(略) S13gdm -> ../init.d/gdm
(略) S20vsftpd -> ../init.d/vsftpd
(略) S91apache2 -> ../init.d/apache2
... (以下略) ...
```

 在 runlevel 2 中移除 Apache2：

```
dbtsai@ubuntu:~ $ sudo rm /etc/rc2.d/S91apache2
```

再次提醒您喔，移除了以後，服務還是在跑，這個動作只不過是讓下次開機預設執行，所以若要關閉服務需要手動執行控制 script 去關閉哩！

 在 runlevel 2 中加入 Apache2：

```
dbtsai@ubuntu:~ $ sudo ln -s /etc/init.d/apache2 /etc/rc2.d/S91apache2
```

加入新的 runlevel 後，也只不過是讓系統下次開機的時候預設執行，所以要馬上啟用，還是得要手動去啟動唷！還有後面的數字最好是和原本的一樣，因為若改的比較前面的話可以會因為其他相依的服務還沒啟用，導致啟動失敗。改太後面的話，就會讓該服務比較晚啟動，有時候會影響後面的服務。所以建議讀者原

來該服務用多少號碼，之後加上去的時候也用原來的吧！當然一般伺服器服務通常是不會遇到這種問題，因為彼此之間是獨立的，但為了謹慎起見，還是用原來若您已經砍了，可以看看在別的 runlevel 是放在哪裡！

使用 update-rc.d 來管理 runlevel

這個程式是 Ubuntu 預設就有安裝的，預設是修改的時候會從 runlevel 0 到 6 都改到！當然也可以指定說只改某一個 runlevel。

在所有 runlevel 中加入 Apache2 ：

```
# 如以下範例，只要在 -f 後面加上服務名稱即可！ defaults 的參數是讓所有
# runlevel 都加入此設定！細心的讀者一定有發現，runlevel 0, 1 和 6 加入的
# 不是 S 啟動指令，而是 K 開頭的關閉指令！這是因為在 runlevel 0 是關機，
# 所以當然要正常的把 Apache 關閉啦！在 runlevel 6 也是同樣的原因。
# runlevel 1 是單人模式，不會啟動伺服器，所以也是 K 開頭的。
#
# 可以用 update-rc.d -f apache2 defaults 91 來指定 K 和 S 後面的數字是 91，
# 也可以用 update-rc.d -f apache2 defaults 91 20 來指定 S 後面是 91，在
# runlevel 0, 1 和 6 裡面的 K 是 20。
#
# 若把 defaults 換成 multiuser 的話，就只會在 runlevel 2 到 5 作更動，
# 也就是不會在 0, 1, 和 6 裡面加入 K 開頭的檔案。
dbtsai@ubuntu:~ $ sudo update-rc.d -f apache2 defaults
Adding system startup for /etc/init.d/apache2 ...
  /etc/rc0.d/K20apache2 -> ../init.d/apache2
  /etc/rc1.d/K20apache2 -> ../init.d/apache2
  /etc/rc6.d/K20apache2 -> ../init.d/apache2
  /etc/rc2.d/S20apache2 -> ../init.d/apache2
  /etc/rc3.d/S20apache2 -> ../init.d/apache2
  /etc/rc4.d/S20apache2 -> ../init.d/apache2
  /etc/rc5.d/S20apache2 -> ../init.d/apache2
```

在所有 runlevel 中移除 Apache2 服務 ：

```
dbtsai@ubuntu:~ $ sudo update-rc.d -f apache2 remove
Removing any system startup links for /etc/init.d/apache2 ...
  /etc/rc0.d/K20apache2
  /etc/rc1.d/K20apache2
```



```

/etc/rc2.d/S91apache2
/etc/rc3.d/S91apache2
/etc/rc4.d/S91apache2
/etc/rc5.d/S91apache2
/etc/rc6.d/K20apache2

```

指定某個 runlevel 中增加 Apache2 啟動或關閉服務：

```

# 以下指令會在 runlevel 2 到 5 產生 S91apache2，當然可以指定只要 runlevel 2，
# 此時只要把 3 4 5 去掉。注意在最後面要加上一個點（.），並且要和前一個數字
# 空一格！
dbtsai@ubuntu:~ $ sudo update-rc.d -f apache2 start 91 2 3 4 5 .
Adding system startup for /etc/init.d/apache2 ...
  /etc/rc2.d/S91apache2 -> ../init.d/apache2
  /etc/rc3.d/S91apache2 -> ../init.d/apache2
  /etc/rc4.d/S91apache2 -> ../init.d/apache2
  /etc/rc5.d/S91apache2 -> ../init.d/apache2

# 若要指定 K 開頭的關閉服務，可以透過以下指令！這個指令會在 runlevel
# 0, 1 和 6 建立 K20apache2 的 soft link。記得注意最後的點。
dbtsai@ubuntu:~ $ sudo update-rc.d -f apache2 stop 20 0 1 6 .
Adding system startup for /etc/init.d/apache2 ...
  /etc/rc0.d/K20apache2 -> ../init.d/apache2
  /etc/rc1.d/K20apache2 -> ../init.d/apache2
  /etc/rc6.d/K20apache2 -> ../init.d/apache2

# 以上兩個指令也可以合併成爲以下一行
dbtsai@ubuntu:~ $ sudo update-rc.d -f apache2 start 91 2 3 4 5 . stop 20 0 1 6 .
Adding system startup for /etc/init.d/apache2 ...
  /etc/rc0.d/K20apache2 -> ../init.d/apache2
  /etc/rc1.d/K20apache2 -> ../init.d/apache2
  /etc/rc6.d/K20apache2 -> ../init.d/apache2
  /etc/rc2.d/S91apache2 -> ../init.d/apache2
  /etc/rc3.d/S91apache2 -> ../init.d/apache2
  /etc/rc4.d/S91apache2 -> ../init.d/apache2
  /etc/rc5.d/S91apache2 -> ../init.d/apache2

```

使用 sysvconfig 來管理 runlevel

首先您需要先安裝這個程式。

✎ 安裝 sysvconfig :

```
dbtsai@ubuntu:~ $ sudo apt-get install sysvconfig
```

✎ 執行 sysvconfig :

```
dbtsai@ubuntu:~ $ sudo sysvconfig
```

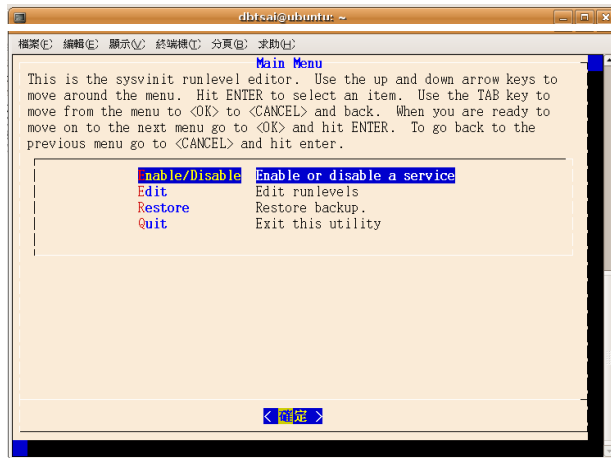


圖 13.2 sysvconfig.png



圖 13.3

執行後會看到如圖 13.2 的畫面。如圖所示，我們先介紹第一個選項的功能 (Enable or disable a service)。第一個選項進去後，會出現 13.3 的畫面，此時讀者可以在您想要啟動的程序上打上 (*)，如此該服務就會在開機的時候啟動。注意喔，這裡所作的更動會同時更動到 runlevel 0 到 6 唷！接下來選擇確定離開此畫面，如下圖選擇 Finished 就完成 runlevel 的更動啦！很簡單吧！

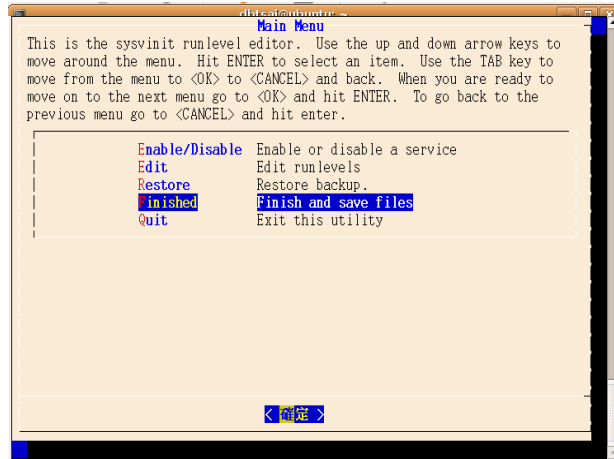


圖 13.4

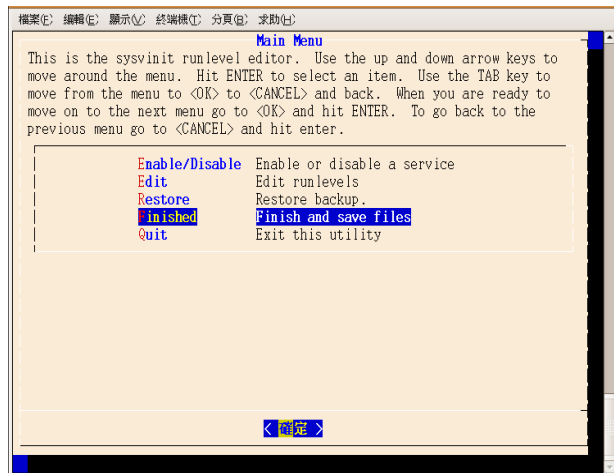


圖 13.5

若要一項一項做細部設定，選擇 Edit runlevels 後，會有如圖 13-3-4 的畫面出現，讀者可以用 delete 鍵把上面的數值清空，接下來打上自己要的值，如 S91 等等，如此該服務就可以開機時預設啟用啦！若您把他保留空白，就不會啟用唷！記得退出這個畫面後，要選擇 Finish 存檔喔。

使用 services-admin 的圖形介面設定 runlevel

這個程式預設在系統內就有內建，讀者可以從選單上系統→管理→服務找到。也可以如以下使用指令啟動。

 執行 services-admin :

```
dbtsai@ubuntu:~ $ sudo services-admin
```

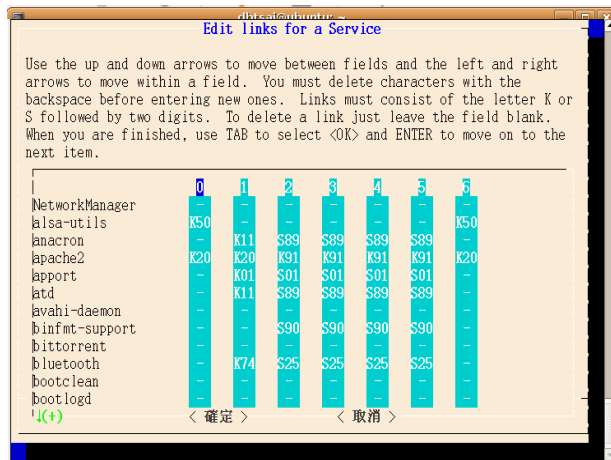


圖 13.6



圖 13.7

執行後會出現圖 13.7，讀者可以在這裡對個別服務設定啟動或不啟動。但是這裡直接勾選是會讓 runlevel 2 到 5 都啟動。因此讀者可以按右鍵，選擇屬性，此時會看到圖 13.7。這裡讀者可以對個別服務設定在某個 runlevel 是否要啟動，以及執行順序。

CHAPTER 14

網頁伺服器（LAMP 架構）

很多人已經把「上網」當成「逛網頁」的代名詞，可見 Web 在整個網路應用是佔很大的一部分。在網站的網址<http://www.emook.org>常常看到起頭是 www，那就是 World Wide Web（全球資訊網）的簡稱哩！所以大家常去<http://www.google.com.tw>或<http://tw.yahoo.com>都是屬於 Web 呢！

隨著時代的進步，從前靜態網頁的技術慢慢不夠用了，靜態網頁缺少了和使用者互動，只能單方面提供資料。於是動態網頁的技術開始蓬勃發展，如討論區，留言版和網路銀行等等，這些網站不只能提供我們資料，還能讓我們輸入資料，並且依照程式設計去處理這些互動的資訊。

傳統的靜態網頁只需要網頁伺服器 Apache 就夠了，它能提供網頁給瀏覽器透過 http 協定觀看。但進步到動態網頁的時候，伺服器必須要處理我們輸入或要求的資訊，故須透過網頁程式語言來處理這些要求，並且把這些資料儲存在資料庫裡面，供未來使用。網頁語言目前有很多種可以選擇，如 PHP、ASP、JSP、Perl 和 Python 等等，各有自己的特色，資料庫可以選擇的也很多，如 MySQL、PostgreSQL、Oracle 和 MS SQL 等等。目前在大部分的中小型網站最流行的是使用 PHP 語言配合 MySQL 使用。PHP + MySQL 的應用程式絕大多數都可以在網路上免費下載的到，例如很多人愛用的 XOOPS 架站機、wikipedia 這套 wiki

軟體和 phpBB 論壇討論區等程式，都是免費的 PHP 大型代表作。Ubuntu 中文站也是用這些軟體去拼湊，根本不需要寫到一行程式碼就可以架出專業的網站。

那什麼是 LAMP 呢？聰明的讀者一定猜到了，就是 Linux + Apache + MySQL + PHP 所組成的動態網頁系統，就稱為 LAMP 啦！這也是目前中小型網站占最多的架站方式。由於這本書的重點在裝好 LAMP 環境，因此裝 XOOOPS 或者其他網頁應用軟體的方式請自己上網查詢，市面上也有很多本書有詳細討論。

14.1 LAMP 網頁伺服器運作原理

網址 <http://www.emook.org> 的 `http` 代表了網頁傳輸協定，它是 Hypertext Transfer Protocol (超文本傳輸協定) 的簡寫，這定義了 Server 和 Client 都需要依照這個遊戲規則來玩，這樣才能確保彼此互通。這個傳輸協定是由 W3C (World Wide Web Consortium) 這個組織所維護的，它們的網址是 <http://www.w3c.org/>。

HTTP 運作原理很簡單，瀏覽器先向 Server 提出要求，在要求裡面有一些瀏覽器給 Server 的表頭，如使用者端的瀏覽器版本、語言和可以接受的壓縮編碼等等，然後 Server 再根據這些要求送出回應表頭和文件內容。注意，Client 端提出來的要求表頭內並不一定會被接受，要看 Server 有沒有支援而定，所以最後瀏覽器接收到內容後還是根據 Server 最後給它的表頭解釋喔！典型的例子是瀏覽器向 Server 說它支援壓縮格式的網頁，但是若 Server 不支援的話，會略過這個要求並且在回送給瀏覽器的表頭上不會有網頁被壓縮的資訊，所以瀏覽器一定是依據 Server 送回來的表頭來判讀資料。當瀏覽器不支援壓縮時，那麼在一開始請求的表頭就不會有壓縮的資訊，若 Server 有支援壓縮的話，此時也不會自作主張的壓縮網頁，所以浏览器和 Server 透過表頭的資訊來溝通是很重要的，在這個步驟，雙方將會取得最大的相容性。

目前在 Linux 上常見的方式是配合以下三套軟體來達成動態網頁的效果。

- **Apache** : 它是目前全球網頁伺服器市場佔有率最高的軟體，目前比微軟的 IIS 佔有率還要高三倍所左右！它是由 NCSA 的 `httpd 1.3` 為基礎，做了大量的修改和 patch，所以有人戲稱這是 A patchy 版，諧音近似 Apache。其實它的命名是從一個印地安部落 Apache 的名稱來的，並且組成基金會

以自由軟體的型式運作。目前在大部分的平台都可以執行，如 MS Windows、Linux、MacOSX 和 FreeBSD 等。

- **PHP**：這是一種直譯式腳本語言，主要應用於伺服器端的應用程式和動態網頁開發。它的命名是從「PHP: Hypertext Preprocessor」的遞歸縮寫而來。它主要會被嵌入在 HTML 上在 Web 伺服器上運行，當程式被瀏覽器請求時，程式會在伺服器上執行，結果會轉成普通的 HTML 文件回傳給 Client 端瀏覽器，所以整個流程在 Client 端是不需要計算的！因為它是直譯式語言，有優秀的跨平台能力，所以幾乎在不同的系統都可以運作一樣的程序碼，造就了各種不同 PHP 應用軟體的蓬勃發展。
- **MySQL**：它是一個開放原始碼的資料庫產品，開發者為 MySQL AB 公司。由於它的效能不錯，且體積小，所以它廣泛的使用在中小型網站中。它允許各種不同的資料放在資料庫中，然後用不同的語言來查詢，雖然功能上沒有大型商業公司開發的 Oracle 和 DB2 等那麼強大，但因為幾乎免費和功能已經足夠中小型網站使用，因此流行起來。

在 LAMP 架構的網站中，當 Client 向 Server 上的 Apache 要求連線後，若 Apache 發現它要求的網頁檔是 PHP 撰寫的時，就會先送到 PHP 直譯器去執行，若程式碼裡面有需要查詢資料庫的地方，就由 PHP 引擎和 MySQL 查詢資料，然後 PHP 把單純的 HTML 結果送回去給 Apache，最後再交給 Client 端的瀏覽器顯示在螢幕上。所以在客戶端是完全沒有執行到程式的，它只不過是把伺服器送來的純網頁顯示到螢幕上而已，所有的計算都是在伺服器上完成，所以伺服器上也可以選用不同的網頁程式語言，如 jsp、perl 或 python 等，完全不會影響到客戶端的結果。

14.2 LAMP 安裝架設

現在要開始示範如何安裝 LAMP，這些套件將會直接用 APT 套件管理系統安裝。因為安全更新問題，筆者非常不建議伺服器相關套件從 tarball 自己編譯安裝。筆者認為從 tarball 安裝的安全性一定會比從套件管理系統安裝來得差，因為套件庫裡的程式都有專人在維護安全性漏洞，從 tarball 裝的話，你要自己常常去看漏洞公告，一般人應該不會那麼勤勞吧！以下筆者將會逐步說明如何安裝 Apache、PHP 和 MySQL。

14.2.1 Apache2 安裝

Apache2 和 Apache1.3 系列最大的變化是效能的改善，而這些改善就是透過 multiple child processes with multiple thread 來達成。雖然在 PHP 的使用環境下建議還是用傳統的 multiple child processes with one thread 會比較穩定，但這裡筆者還是稍微介紹一下其他 thread models！在 Apache2 有支援三種 thread models，這些模組稱為 Multi-Processing Modules (MPMs)。

- **apache2-mpm-worker**：這是 Ubuntu 預設會安裝的套件，它運作的時候會有數個 child processes，每個 processes 都會再有數個 threads。由於同一個 processes 內的不同 thread 都可以共用記憶體資料，所以可以減少記憶使用並且增加效能！但是很多第三方的套件沒有 thread safe，也就是沒有考慮到 muti-thread 的問題，所以若跑在 worker thread model 的 Apache 會很不穩定！例如 PHP 就不支援 thread safe。
- **apache2-mpm-prefork**：這是最傳統的模式，運作時會有數個 child processes，每個 processes 都只跑一個 thread。因此每個 thread 都隸屬於不同 processes 不會互相干擾，雖然效能較差且耗費比較多記憶體，但是對於有安裝一些第三方的模組時穩定性比較好！例如安裝 PHP 時一定要用這個 thread model，不過若先裝 Apache 預設的 mpm-worker 也沒關係，偉大的套件管理系統會自動幫你換成 mpm-prefork 版的 Apache。
- **apache2-mpm-prechild**：這個和 mpm-worker 有點像，不過它允許不同的 processes 用不同的使用者執行，目前這還在開發中，非常不穩定！

介紹完了以後，我們可以透過 metapackage 虛擬套件把 apache2 很簡單的裝起來吧！

安裝 Apache2：

```
sudo apt-get install apache2
```

若讀者想要明確的指定要安裝 mpm-prefork 版而不要預設的 mpm-worker 版本，可以透過以下指令安裝。這個是非必要的唷，因為在裝 PHP 的時候會自動把 worker 換成 prefork 版！

指定安裝 mpm-prefork 版 Apache2 :

```
sudo apt-get install apache2 apache2-mpm-prefork
```

若要關閉或重新啟動 apache 的話，可以透過以下控制 script 達成。

關閉 Apache2 服務或重新啟動 :

```
sudo /etc/init.d/apache2 stop # 參數可以是 restart 或 start 等
```

讀者在啟動的時候若遇到以下錯誤訊息，可能是/etc/hosts 沒有設定好！讀者可以參考 6.1.6 節關於 hostname 和 hosts 的部份。

hostname 設定錯誤的警告訊息 :

```
* Starting apache 2.0 web server... apache2: Could not determine the server's
fully qualified domain name, using 127.0.1.1 for ServerName
```

雖然有以上錯誤但 apache 還是可以使用，但是可能會在虛擬主機的部份出現問題。若正確設定好時，hostname 應該顯示該主機的名稱，hostname -f 會顯示該主機的完整 FQDN 名稱，也就是主機名稱加域名。

hostname 設定錯誤的警告訊息 :

```
dbtsai@ubuntu:~$ hostname
www
dbtsai@ubuntu:~$ hostname -f
www.emook.org
```

有關於開機後要不要自動啟動 Apache 是屬於 runlevel 的問題，讀者可以參考第十三章關於 runlevel 的討論。

在安裝好 Apache 後，讀者可以試著用瀏覽器連線到主機試試看吧！看能不能正常連線，詳細測試我們等到安裝完 PHP 在一起測試。

14.2.2 PHP 安裝

目前在 Ubuntu 裡面有提供 PHP4 或者 PHP5 兩種不同的系列，由於某些舊的程式沒辦法在 PHP5 裡面順利執行，所以讀者需要自己視情況選擇。基本上 PHP5

的執行效率會比較好，且目前大部份套裝 PHP 軟體都可以在 PHP5 裡面執行，若讀者有自己開發的程式無法在 PHP5 裡面執行，那麼就需要安裝舊版的 PHP4。

表 14-2-1 PHP5 重要的相關套件

套件名稱	功能
php5	安裝 PHP5 一些必要套件的 metapackage
libapache2-mod-php5	Apache2 的 PHP5 模組
php5-gd	PHP5 的圖形相關模組，常用於相簿這類的程式
php5-mysql	PHP5 連線到 MySQL 資料庫的模組
php5-mysqli	同上，這個效能比較好，但是有用到比較新的 API，所以 MySQL Server 版本要大於 4.1。因為 Ubuntu 的 MySQL 已經是 5.0 了，所以安裝這一個吧！
php5-cli	很多 PHP 魔人覺得寫 PHP 比 bash shell script 簡單多了，安裝這個，就允許你在 shell script 裡面使用 PHP 的語法唷！

如表 14-2-1，這些是常用的 php5 套件，若讀者還有其他的需求，用 `apt-cache search php5` 找一下吧！

安裝 PHP5 :

```
sudo apt-get install php5 libapache2-mod-php5 php5-gd php5-mysqli
```

安裝 PHP4 的方式這裡就不再贅述了，把 5 改成 4 就可以啦！安裝完以後，記得重新啟動 Apache 唷，這樣才會把 PHP 模組載入到系統裡面。

重新啟動 Apache :

```
sudo /etc/init.d/apache2 restart
```

由於 Apache 預設放置網頁的位置是 `/var/www` 裡面，所以我們寫個簡單的 PHP 測試程式來看看 PHP 是否正確執行。這個程式將會顯示 PHP 和 Apache 的詳細資訊。

編輯 `/var/www/phpinfo.php` :

```
<? phpinfo(); ?>
```

接下來用瀏覽器連線到你 server 上的 phpinfo.php 吧！讀者應該會看到如圖 14-2-1 的畫面，這代表了讀者的 Apache 和 PHP 已經正常運作了唷！

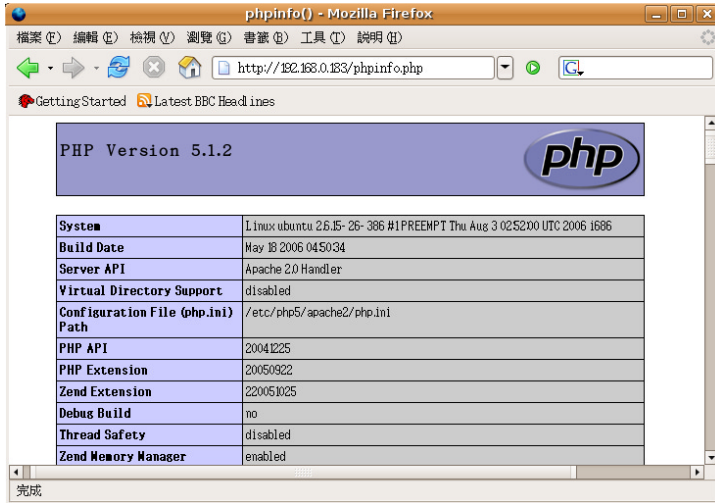


圖 14-2-1

確認正常的話，趕快把 phpinfo.php 刪除，裡面有一些你主機的組態設定，可能會讓駭客掌握到而對安全性有所影響哩。


14.2.3 MySQL 安裝

安裝 MySQL 很簡單，先用以下指令安裝吧！

 安裝 MySQL：

```
sudo apt-get install mysql-server mysql-client
```


MySQL 所使用的密碼和系統上的是分開的，而安裝好後預設的 MySQL root 密碼是空的，所以我們第一件事情就是要修改 MySQL root 密碼，不然會很危險唷！

 修改 MySQL 密碼：

```
sudo mysqladmin -u root -p password <輸入你要的新密碼>
```

修改密碼的時候，它會問你原來的密碼，此時你按 ENTER 跳過去即可，因為預設密碼是空白的！

若你需要重新啟動 MySQL 的話，可以使用以下指令。

 重新啟動 MySQL：

```
sudo /etc/init.d/mysql restart
```

若要讓它不要再開機的時候啟動，也是修改 runlevel 啦！讀者自己去參考一下相關資訊吧！

14.3 Apache2 設定

Debian 裡面的 Apache2 和其他發行版的管理方式有點不同，在 Debian 的設定檔不再是傳統的 httpd.conf，而是放在 apache2.conf，並且把很多模組設定或者虛擬主機設定拆開來放到各自的目錄。這些獨立出來的設定檔又可以透過 Debian 專屬的工具程式啟用或關閉，雖然這樣的設計對於初學者會覺得比較困難，但是若你在規劃大型網站時會發現這樣模組化的管理其實是比較方便的，而 Ubuntu 也繼承到這個特性。

所以筆者將會先介紹整個系統架構的輪廓，然後慢慢的澄清細部的設定，希望讀者能瞭解到 Ubuntu 裡面 Apache2 管理精髓。

14.3.1 系統架構


在其他的發行版中，一般是把所有的設定包含虛擬主機，啟動時載入的模組寫到 httpd.conf 裡面。在 Ubuntu 中把大部分的 httpd.conf 的東西移到 apache2.conf 裡面，且把虛擬主機和模組設定又獨立出來。可用虛擬主機的設定放在 /etc/apache2/sites-available/ 裡，這些設定其實是不啟用的，需要做一個符號連結到 /etc/apache2/sites-enabled/ 才算是正式啟用。同理，可用模組相關設定放在 /etc/apache2/mods-available/ 裡，實際要啟用還是要弄一個 soft link 到 /etc/apache2/mods-enabled/。

讀者可以想像成類似 runlevel 的概念，在開機的時候跑到 runlevel 裡面的某一個啟動程式，其實是 soft link 到 /etc/init.d 裡面的控制檔，因此我們可以很容易決定開機的時候要跑哪些服務。在 Apache2 使用類似方法的好處是我們可以修改 sites-available 來增加新的虛擬主機，然後手動 soft link 到 sites-enabled 來啟用。當那天要關閉該虛擬主機時，只要砍了 sites-enabled 裡面的 link 就可以關閉了，原來的設定檔不需要動到，若之後想要回復就只需要再做一次 soft link。而模組啟動也是一樣的原理，此時可以先在 mods-available 裡面寫好不同的模組設定，但是不啟用。若真的要啟用這些模組，再 soft link 到 mods-enabled 來啟用。

類似於 runlevel 中提供的 update-rc.d 來自動處理這些符號連結的工作，在 Apache2 中也有提供一些工具程式來處理這些模組或站台啟用或關閉，不需要再自己用 `ln -s /etc/apache2/mods-available/php5.load /etc/apache2/mods-enabled/php5.load` 啟用。接下來筆者會先介紹模組啟用方式再介紹虛擬主機啟用方式。

最後筆者會稍微介紹一下 Apache2 其他設定檔在作什麼，讓讀者在本節看完後就可以抓到 Apache 架構的概念，這些設定檔的細節就會在後面慢慢的補上來。

啟動或關閉 mods-available 裡面的模組

 查看可用模組與啟用模組：

```
dbtsai@ubuntu:~$ ls /etc/apache2/mods-available/
auth_dbm.load      deflate.load       ssl.conf          ssl.load
auth_ldap.load     expires.load      php5.conf        suexec.load
cache.load         ext_filter.load   php5.load        unique_id.load
... (略) ...

dbtsai@ubuntu:~$ ls -l /etc/apache2/mods-enabled/
總計 0
...略...      cgi.load -> /etc/apache2/mods-available/cgi.load
...略...      php5.conf -> /etc/apache2/mods-available/php5.conf
...略...      php5.load -> /etc/apache2/mods-available/php5.load
...略...      userdir.conf -> /etc/apache2/mods-available/userdir.conf
...略...      userdir.load -> /etc/apache2/mods-available/userdir.load
```

如以上範例，我們可以看到系統內有其他額外的 Apache2 模組可以使用，且在安裝後預設啟用 PHP、用戶個人網頁和 cgi 模組。注意到這些啟用的模組只不過是個符號連結檔到 mods-available 裡面，讀者只要把它刪除就關閉使用。若是該模組有 load 和 conf 兩個檔，前者是負責讀入該模組的函式庫，後者是該模組的設定檔，所以啟用或刪除時要兩個都建立或刪除符號連結。

關閉 PHP 模組範例 (手工刪除符號連結法) :

```
sudo rm /etc/apache2/mods-enabled/php5.load
sudo rm /etc/apache2/mods-enabled/php5.conf
sudo /etc/init.d/apache2 restart
```

啟用 PHP 模組範例 (手工建立符號連結法) :

```
sudo ln -s /etc/apache2/mods-available/php5.load
/etc/apache2/mods-enabled/php5.load
sudo ln -s /etc/apache2/mods-available/php5.conf
/etc/apache2/mods-enabled/php5.conf
sudo /etc/init.d/apache2 restart
```

記得喔，在關閉或啟用模組後，要重新啟動 Apache 喔！而如同之前提到的，在 Ubuntu 裡面還有工具程式可以自動幫你管理模組開關的符號連結設定，我們現在來試試看吧！

關閉 PHP 模組範例 (使用工具程式) :

```
sudo a2dismod php5
sudo /etc/init.d/apache2 restart
```


啟用 PHP 模組範例 (使用工具程式) :

```
sudo a2enmod php5
sudo /etc/init.d/apache2 restart
```

而筆者比較喜歡透過工具程式來啟用或關閉模組，至少可以少打很多字，也避免打錯字的問題唷！

啟動或關閉 sites-available 裡面的虛擬站台

在 Apache2 安裝好以後，系統裡面只有一個虛擬站台，也就是預設根目錄在 /var/www 的那個網站。在 Debian 和 Ubuntu 裡面，基本上是把所有的站台都當成虛擬網站來看，所以管理上會有比較好的一致性。而以下的範例是筆者伺服器上面的，除了預設的站台以外，又多了一個 www.emook.org.tw 虛擬主機，這樣同一個 Apache Server 就能用同一個 IP 服務數個網站。這裡的重點在於管理站台是否啟動。而範例中的第二個虛擬站台 emook 是筆者自己建立的唷，詳細設定方式，會在後續章節提及。

 查看可用站台與啟用的站台：

```
dbtsai@ubuntu:~$ ls /etc/apache2/sites-available/
default  emook
root@www:/home/dbtsai# ls -l /etc/apache2/sites-enabled/
total 0
...略... 000-cqis -> /etc/apache2/sites-available/default
...略...  emook -> /etc/apache2/sites-available/emook
```

如以上範例，我們看到的架構和符號連結其實和之前講模組啟用與關閉一樣，所以讀者可以使用手工建立或刪除符號連結，或者用 Ubuntu 內附的工具程式。有時候在 sites-enabled 裡的檔案前會有數字，那個沒有關係。我們在工具程式裡面所加的參數名稱是 sites-available 裡面那一些喔！

 關閉 emook 站台範例：

```
sudo a2dissite emook
sudo /etc/init.d/apache2 restart
```


 啟用 emook 站台範例：

```
sudo a2ensite emook
sudo /etc/init.d/apache2 restart
```

重要的設定檔

讀者或許會很好奇，在 Ubuntu 裡面虛擬主機或模組的設定可以那麼模組化，把不同功能的設定寫在不同檔案，再 include 進來，避免了所有設定都放在同一個檔案可以會很亂的問題，也改善了虛擬主機增加或減少的不方便。其實 Ubuntu

的 Apache 和其他的發行版一樣，在啟動的時候是去抓 `/etc/apache2/apache2.conf`，但特別的是，在 Ubuntu 和 Debian 的 `apache2.conf` 如以下內容，有特殊的地方。

 `apache2.conf` 裡面達成設定功能分離的方法：

```

... (略) ...
# 讀入 mods-enabled 裡面所有副檔名是 load 和 conf 的設定檔。
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
... (略) ...
# 讀入 httpd.conf 達成和舊版 apache 的相容性，並讀入 ports.conf 來指定
# Apache2 所使用的通訊埠。
Include /etc/apache2/httpd.conf
Include /etc/apache2/ports.conf
... (略) ...
# 讀入使用者額外設定的設定檔，那麼就不用寫在 apache2.conf，會比較整齊。
Include /etc/apache2/conf.d/[^.#]*
... (略) ...
# 讀入 sites-enabled 裡面所有虛擬主機設定檔。
Include /etc/apache2/sites-enabled/[^.#]*

```

所以在讀取 `apache2.conf` 時，Apache2 會引入 `mods-enabled` 和 `sites-enabled` 以及 `httpd.conf` 和 `ports.conf` 到系統內，這也是為什麼在 Ubuntu 可以做到設定檔功能分離的原因。

表 14-3-1 Apache2 重要的設定檔與目錄

設定檔或目錄	功能
<code>/etc/apache2/apache2.conf</code>	主要的設定檔，包含了系統整體的重要設定，並在裡面引入其他設定檔。
<code>/etc/apache2/httpd.conf</code>	為了早期第三方模組的相容性，這個設定檔提供給那些模組用 <code>apxs2</code> 來作設定。如 <code>FrontPage Server</code> 就會把設定寫在這裡。
<code>/etc/apache2/mods-available/</code>	所有可以使用的模組設定檔都會放在這裡面，其中副檔名是 <code>load</code> 的負責載入函式庫，副檔名為 <code>conf</code> 是該模組的設定檔。


設定檔或目錄	功能
/etc/apache2/mods-enabled/	由於 apache2.conf 只 include 這個目錄裡面所有的 conf 和 load 檔，因此真正要啟用該模組的話，必須從 mods-available 建立符號連結到此目錄。注意在建立符號連結時，若有 conf 檔也需要同時和 load 檔都建立符號連結。當然讀者可以使用 a2enmod 和 a2dismod 來管理。
/etc/apache2/sites-available/	所有可以使用的虛擬主機設定檔放置處，讀者可以在這裡增加你的虛擬主機。
/etc/apache2/sites-enabled/	實際上 apache2.conf 會啟用該目錄底下所有設定，因此讀者需要從 sites-available 建立符號連結到此目錄。也可以使用 a2ensite 和 a2dissite 管理。
/etc/apache2/conf.d/	裡面可以放一些讀者自己的額外設定，會比較容易管理。
/etc/apache2/ports.conf	定義 Apache2 Server 所使用的 port，預設是 80。
/etc/logrotate.d/apache2	紀錄檔循環設定，當流量很大時，可以修改來減少保存期限。
/var/log/apache2/	該目錄底下存放了 Apache 的各種紀錄檔。
/var/www	Apache2 預設的網頁放置處。

表 14-3-2 Apache2 重要的執行檔

指令	功能
/etc/init.d/apache2	控制 Apache 執行，或關閉的 script。後面參數可以加 start，stop 或 restart。
a2enmod	啟動參數所加的模組，原理是建立符號連結到 mods-enable。
a2dismod	關閉參數所加的模組，原理是刪除符號連結。
a2ensite	啟用參數所加的虛擬主機，原理是建立符號連結到 sites-enable。
a2dissite	關閉參數所加的虛擬主機。
apache2ctl -t	檢查設定檔是否有錯誤
apache2ctl -S	列出目前伺服器上所有虛擬主機

14.3.2 apache2.conf 設定

這裡筆者精選在 apache2.conf 裡面比較重要的地方，告訴讀者該設定的意義和注意事項。在該設定檔中，有很多被註解起來的地方實際上已經分離到其他地方了，所以若沒有必要的話，儘量不要把那些註解起來的東西啟用唷！

 apache2.conf 裡面重要的地方：

```
# Apache Server 設定檔放置處
ServerRoot "/etc/apache2"

# Client 多久沒回應就 timeout，網路狀態不好可改大一點。
Timeout 300

# KeepAlive 為是否開啓連續請求的功能，當打開時每次 Client 端建立連線後，
# 可以要求數個檔案，而不必要重新建立連線。建議把它打開，會有比較好的效能。
# 而 MaxKeepAliveRequests 是每次 Client 端建立連線後，可以要求最大的檔案數，
# 讀者設成 0 為不限制，或者可以把它設高一點，尤其是在網頁很複雜的時候，
# 效能會有明顯改善。通常設 100 到 300 之間已經很夠用啦！
# KeepAliveTimeout 是當連續請求時，Client 端多久沒有在繼續請求就把他踢下線。
# 這個單位是秒鐘，不要開太大，不然資源可以被這些不請求的 Client 端佔著。
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

# StartServers 為 Apache2 啟動時，會啓用幾個 processes，讀者可以透過 ps -aux
# 來驗證。若你的網站流量很大，可以改大一點，但是若用量沒有那麼大，會浪費
# 系統資源，通常預設的 5 已經很夠用了。
# MinSpareServers 可閒置的最少 processes，通常都會設定的和 StartServers 一樣。
# MaxSpareServers 最大可閒置的 processes，若你的網站流量時大時小，那麼這個值
# 可以設大一點，避免人多的時候要開新的 processes，但是設太大會增加系統資源。
# MaxClients 最多可以有幾個客戶端連線，預設的 20 似乎有點少，可以視你網站的
# 規模改到 200 以上。但是預設不能超過 256，若要超過的話需要自己額外加上
# ServerLimit 的參數，定義最大可以有幾個 Client，這個值最大到 20000，若
# 還要在大的話就要修改原始碼的定義。
# MaxRequestsPerChild 每一個 processes 可以有多少個連線數要求。通常設成 0 為
# 不限制。
<IfModule prefork.c>
StartServers          5
```

```

MinSpareServers      5
MaxSpareServers      10
# ServerLimit        3000  # 若MaxClient 要大於 256，需要有這一個參數。
MaxClients           20
MaxRequestsPerChild  0
</IfModule>

# 在設定檔內，還有其他 MPM 可以修改，如 worker 和 perchild 等，但這些在有
# PHP 的環境下不會使用到，所以讀者可以不要管他們。
# Apache2 Server 執行時所用的帳號和群組，若沒有特別的理由，可以不用更動。
User www-data
Group www-data


# 全域的錯誤紀錄檔放置處，個別虛擬主機的設定可以在 sites-available 另外指定。
ErrorLog /var/log/apache2/error.log
# 拜託，AddDefaultCharset 千萬不要加上去。有些書會教這個要改成 Big5 來
# 解決亂碼的問題，這是不好的方式！這個方式會在表頭就送出說這個網頁是用什麼
# 語言編碼，但是若你的網頁有 big5 和 utf-8 混用的話，就會造成錯誤！
# 正確的作法是在網頁 html 內都指定編碼，如
# <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
# 後面那些 AddCharset 是指可以支援的編碼，預設都設好了，不要改動。
#AddDefaultCharset  Big5
AddCharset utf-8      .utf8
AddCharset big5       .big5 .b5

# 若沒有明確指定網頁檔名，Apache2 會依序以下順序來嘗試送給 client 端瀏覽器
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml

```

14.3.3 網站設定檔

在 `apache2.conf` 只作一些全域的設定，若要設定個別虛擬主機或網站的設定，需要修改的地方就在 `/etc/apache2/sites-available/` 裡面。這裡筆者先介紹一下預設的網站 `default` 裡面的架構，額外增加的虛擬主機設定方式是一樣的。

 預設網頁 `/etc/apache2/sites-available/default` 的設定檔擷選：

```

NameVirtualHost *
<VirtualHost *>

```

```

ServerAdmin dbtsai@localhost      # 網頁管理者 E-Mail
DocumentRoot /var/www            # 網頁放置的根目錄

# Options 裡面可以有許多參數，以下將個別簡介。
#   Indexes                當 Client 找不到 index.html 等首頁時，就列出該目錄底下
#                           所有檔案。因此有一點危險性存在，通常是建議關閉。
#   FollowSymLinks         讓你在該目錄底下可以讀取到符號連結到非網頁根目錄內的
#                           檔案。例如現在根目錄是 /var/www，我們建立一個從
#                           /home/web 建立一個 soft link 到 /var/www/web 後，
#                           若有開啓這個選項，才能讀取到 /var/www/web 的內容。
#   MultiViews             提供同一個網址有多國語言的功能。原來是依照 Client 送來的
#                           語言偏好 header 送回不同語言版本的網頁。例如 Google
#                           就是採用這種方式達成同一網址有多國語言的功能。
#
# AllowOverride 和 .htaccess 有關。 .htaccess 可以在網頁目錄下取代全域的
# 網站設定，這樣的好處是可以讓一般使用者不用有 root 權限就可以修改
# 參數，使用彈性也會比較大。它可以設定的參數有以下幾個。
#   AuthConfig            可以在 .htaccess 裡面設定網頁認證的功能
#   Options               可以在 .htaccess 裡面重設 Options 選項
#   All                   以上所有功能都能在 .htaccess 被重設
#   None                  不使用 .htaccess 來重設變數

<Directory /var/www/>           # 定義該目錄的網頁權限
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
# 設定 CGI 程式相關參數
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
# 錯誤紀錄檔放置的地方
ErrorLog /var/log/apache2/error.log
# 錯誤紀錄檔的紀錄層級，不同的設定會有不同的紀錄精細度。

```

```

LogLevel warn
# 瀏覽紀錄檔的放置處，裡面紀錄了Client 端詳細資訊
    CustomLog /var/log/apache2/access.log combined
    ServerSignature On
</VirtualHost>

```


14.3.4 增加虛擬主機

虛擬主機 Virtual Host 的定義是一台 Apache 主機可以處理超過一個 domain，例如有兩個網址都同時對應到該主機 IP，而主機能依照網址不同給與不同的網頁。這種技巧用最多的是主機出租商，因為他們不可能給每個承租的人個別主機與 IP，所以用這種方式在同一台主機上架很多網站。

實作的方式有兩種，一種是 IP Based，另外一種是 Host Based。所謂的 IP Based 是該主機有很多的 IP，用戶連結到不同 IP 會有不同的結果，但是這種方式成本很高，尤其是在目前 IP 短缺的情況下，但優點是對用戶瀏覽端相容性最好。Host Based 是不同網域對應到同一個 IP，連線到這些不同的網域會有不同的結果。讀者或許會奇怪，實際連線不是和網域沒關，那怎樣達成這種功能呢？記不記得 Client 和 Server 溝通時，在 HTTP 1.1 協定中規定 Client 要在表頭送出 Host 告訴 Server 是要連到哪一個虛擬主機，因此 Server 可以依據這個判定要送回網頁是哪一個。因此使用 Host Based 的技術必須要瀏覽器有支援才行，目前幾乎市面上所有瀏覽器都有支援了，故現在主流使用的技術是 Host Based。

讀者若要建立新的虛擬主機，首先要確定的是該網域已經設定好 DNS 紀錄，並將該紀錄指向主機的 IP。筆者的環境是虛擬主機的網址是 www.emook.org，而虛擬主機網頁放置的地方是 `/home/emook`。

大部分虛擬主機的需求是 Host Based，所以這裡我們示範的是 Host Based 的 Virtual Hosting。由於預設主機的設定檔 default 已經是很完整的架構，我們就直接用它來當範本稍微修改給虛擬主機用吧！

 拷被範本：

```

cd /etc/apache2/sites-available
sudo cp default emook # 這個 emook 是筆者自己取的，讀者可以自己命名

```

接下來編輯這個新的設定檔。

新的虛擬主機 /etc/apache2/sites-enabled/emook 設定檔擷選：

```
# NameVirtualHost *                                # 這一行要註解起來

    DocumentRoot /home/emook                        # 修改網頁放置的根目錄
    ServerName www.emook.org                        # 加入該虛擬主機所傾聽的網域
    <Directory /home/emook>                          # 修改定義網頁權限的路徑
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
```

簡單的說，只有要改 4 個地方，先註解掉 NameVirtualHost 後，再修改兩個和根目錄有關的路徑，還有加入新設定 ServerName，其他的地方都可以不需要更動。

啟用新的虛擬主機站台：

```
sudo a2ensite emook
/etc/init.d/apache2 restart
```

14.3.5 使用 mod_deflate 來壓縮網頁並節省頻寬

要加快網頁讀取速度，可以從擴充網站頻寬或者瘦身網頁著手。但在台灣頻寬很貴的情形下，如何節省網站所使用的頻寬來容納更多使用者上線是更重要的議題。如大家所知道的，純文字檔可以壓縮到非常小，大概是原來的 40% 以下，若是你的網站以文字為主的話，啟用網頁壓縮大概可以節省一半以上的頻寬。壓縮網頁的好處不只是你網站節省一半的頻寬，讓可以同時上站的人數成為兩倍，也因為網頁壓縮順便讓用戶更快瀏覽網頁呢！當然壓縮網頁也是有壞處的，它會增加伺服器的 CPU 處理負擔，不過在衡量頻寬和 CPU 處理時間，算一算還是壓縮網頁比較划算。目前這種技巧在各大網站都有使用，如 Google 和 Yahoo 等。

或許讀者會怕會不會有一些瀏覽器不支援，那送過去壓縮的網頁會不會讓他們不能看呢？其實不用擔心，因為在瀏覽器和伺服器請求網頁的表頭，會告訴伺服器可以接受的壓縮格式，因此當瀏覽器沒有支援壓縮時，伺服器不會送出經過壓

縮的網頁，況且目前市面上幾乎所有主流的瀏覽器都已經支援網頁壓縮的功能，因此筆者是建議大家都可以開啟網頁壓縮模組！

若讀者想要知道某個網頁伺服器是否有支援網頁壓縮，必須要從瀏覽器和伺服器的溝通表頭判讀，但是因為預設 Firefox 沒有支援讓使用者知道這些訊息，所以需要安裝一套 livehttpheaders <http://livehttpheaders.mozdev.org> 的外掛，若你只是要啟動壓縮模組但不想知道哪些網站有提供壓縮功能，這個可以不用安裝。安裝後，連到你想要測試的網頁，在 Firefox 裡面選擇「工具」→「頁面資訊」→「header」可以看到如圖 14-3-1 內顯示的表頭資訊。在標記 1 的地方是瀏覽器傳給伺服器的支援壓縮格式，標記 2 是伺服器告訴瀏覽器所使用的壓縮格式。當瀏覽器有支援，但伺服器沒有支援時，那麼在標記 1 個地方還是一樣，但是標記 2 的地方就不會有壓縮資訊，因此瀏覽器就會用沒有壓縮的方式處理。這個外掛同時也能得知網頁的大小，讀者可以試著啟用壓縮和沒有啟用壓縮來比較看看，在筆者的網頁中，首頁沒有壓縮有 6k 左右，壓縮後大約只剩下 1.7k，大約變成原來的三分之一。

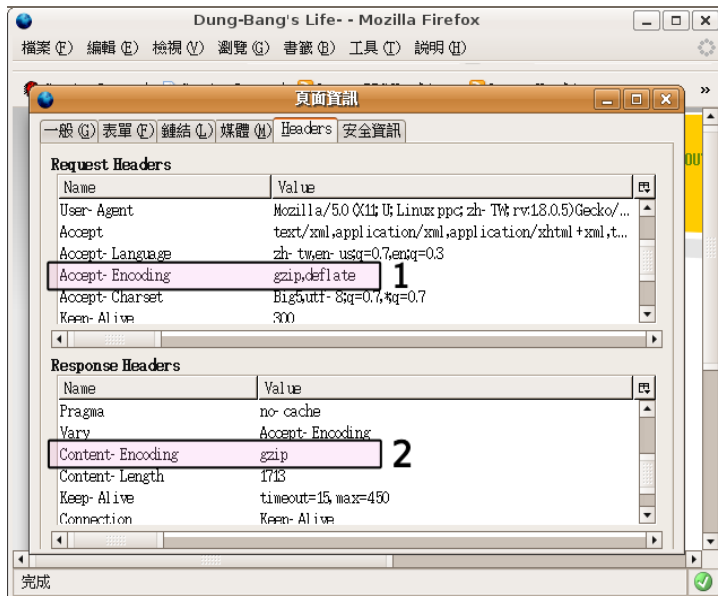


圖 14-3-1 判別某網頁伺服器是否有啟動壓縮的功能

 啟用網頁壓縮模組：

```
sudo a2enmod deflate
```

當啟用了壓縮模組後，預設網頁還是不會壓縮，必須要自己指定哪些類型的檔案要壓縮。這個設定可以放在各個虛擬主機的個別設定內 (sites-available)，讓不同的主機有不同的規則，或者自己寫一個新的設定檔放在 conf.d 裡面，這個就讓讀者視情形自己決定。筆者自己的習慣是放在 conf.d 裡面讓所有的虛擬主機可以享用到網頁壓縮的好處。

編輯壓縮模組設定檔 /etc/apache2/conf.d/deflate.conf :

```
<IfModule mod_deflate.c>
# 壓縮率設定。值可以是 1 到 9，值越大壓縮率越高，網頁越小。
# 通常設為 5 或 6 就夠了，再高就是浪費系統資源。
DeflateCompressionLevel 6

# 由於圖檔已經有壓縮過，再壓縮效率不好，或者如 MP3 等檔案，再壓縮效率
# 不高，所以我們明確的指定哪一些格式要壓縮！格式列表讀者可以參考
# /etc/mime.types。有些人使用全部壓縮的設定，在另外排除不要壓縮的檔案，
# 不過這樣很容易有忘了排除的，導致壓縮到非文字檔使 Server 效能變差，所以
# 筆者還是喜歡使用個別指定的方式。以下是筆者所用的內容，讀者可以參考使用。
AddOutputFilterByType DEFLATE text/plain
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE text/css
AddOutputFilterByType DEFLATE application/xhtml+xml
AddOutputFilterByType DEFLATE application/xml
AddOutputFilterByType DEFLATE application/rss+xml
AddOutputFilterByType DEFLATE application/atom+xml
AddOutputFilterByType DEFLATE application/x-javascript
AddOutputFilterByType DEFLATE application/x-httpd-php
AddOutputFilterByType DEFLATE application/x-httpd-fastphp
AddOutputFilterByType DEFLATE application/x-httpd-eruby
AddOutputFilterByType DEFLATE image/svg+xml
AddOutputFilterByType DEFLATE application/postscript

<IfModule mod_headers.c>
# 處理壓縮過網頁經過 proxy 的問題
Header append Vary User-Agent
</IfModule>
</IfModule>
```

這裡再次提醒讀者，網路上有些人使用的規則是先設所有檔案都要壓縮，然後再把不要壓縮的另外設出來。此時若你例外沒有設到 avi 檔，這時候當你的網站有開放這類的檔案讓人下載，你就慘啦！這種檔案壓縮都要幾十分鐘，耗費 CPU 資源很大，壓起來幾乎不會變小，此時你的網站反而會像龜速！所以建議你只壓縮文字檔格式的檔案。

重新啟動 Apache2 :

```
sudo /etc/init.d/apache2 restart
```

讀者可以透過之前討論的 Firefox 外掛 livehttpheaders 來確認你的網頁主機是否已經啟用壓縮機制。讀者也可以使用一些流量監控軟體，如 MRTG 等等，你會發現你的流量變小大概一半唷！使用者也會減少抱怨網路很慢啦！

讀者可以把各個使用者讀取網頁的壓縮率寫到紀錄檔中，之後也可以用一些工具程式來分析網站壓縮後流量減少多少，和壓縮策略等進階議題。

由於這個紀錄檔的設定檔要寫在各個虛擬主機設定裡面，無法放在全域變數裡，但是把它直接寫在 sites-available 裡面會很亂，所以筆者是習慣額外寫在其他地方，在用 include 的方式引入。這樣測試完成後，也可以取消這個紀錄檔紀錄的功能，來節省系統檔案空間。

編輯 /etc/apache2/DeflateNote.conf :

```
<IfModule mod_deflate.c>
    DeflateFilterNote Input instream
    DeflateFilterNote Output outstream
    DeflateFilterNote Ratio ratio
    LogFormat "%r" %{outstream}n/%{instream}n (%{ratio}n%%)' deflate


# 紀錄檔位置設定建議不要放在這裡，這樣可以多個虛擬主機共用 DeflateNote.conf
# 並且紀錄檔都分別紀錄到不同的地方。
# CustomLog /var/log/apache2/deflate.log deflate
</IfModule>
```

接下來要把這個功能放到虛擬主機設定裡面，編輯你要紀錄的虛擬主機設定檔吧！

 編輯 /etc/apache2/sites-available/default :

```
<VirtualHost *>
# 在VirtualHost 後面加上以下這兩行。
include /etc/apache2/DeflateNote.conf # 引入壓縮率紀錄
CustomLog /var/log/apache2/deflate-default.log deflate # 設定紀錄檔放置位置
```

設定好後，重新啟動 Apache2，最後開放主機給使用者測試，過一段時間後，讀者可以去分析紀錄檔，看省下來的頻寬有多少，各種不同格式的壓縮率是多少，是否有可以壓縮的檔案類別沒有加上或不可壓縮的類別被壓縮到了，再去細部調整規則。


 觀看壓縮率紀錄檔：

```
dbtsai@ubuntu:~ $ more /var/log/apache2/deflate-default.log
"GET /cindex.php HTTP/1.1" 5558/26844 (20%)
"GET /pic/pic%20dll/c-paper1.gif HTTP/1.1" -/- (-%)
"GET /cindex.php?name=./c/project.htm HTTP/1.1" 9240/30380 (30%)
"GET /pic/title/c-program.gif HTTP/1.1" -/- (-%)
"GET /cindex.php?name=links.htm HTTP/1.1" 5281/24956 (21%)
"GET /pic/title/tlink.png HTTP/1.1" -/- (-%)
"GET /pic/lorp.gif HTTP/1.1" -/- (-%)
```


如筆者網站，幾乎所有純文字檔的網頁壓縮後只有原來的 25% 左右，若考慮圖檔沒有壓縮的話，大概頻寬可以節省一半，並且 CPU 資源大約多個 5% 而已。使用者也回報網站反應速度變快了！記得喔，測試完了以後，建議把壓縮率紀錄檔拿掉，不然很佔空間。

14.3.6 啟用個人網頁

在 Ubuntu 裡面，Apache2 把使用者個人網頁獨立到一個額外的模組和設定檔。因此要開啟或關閉使用個人網頁的功能是很簡單的。


 啟用個人網頁：

```
sudo a2enmod userdir
sudo /etc/init.d/apache2 restart
```

 關閉個人網頁：

```
sudo a2dismod userdir
sudo /etc/init.d/apache2 restart
```

個人網頁的設定一般可以不需要修改，但是預設的設定有啟用 Indexes，也就是瀏覽器在找不到預設網頁時會列出所有該目錄下的檔案，所以讀者可以視自己需要去修改預設的設定。這些設定的內容已經在 14-3-2 有詳細的說明，讀者可以參考那裡的解釋。

 個人網頁設定檔 /etc/apache2/mods-available/userdir.conf：

```
<IfModule mod_userdir.c>
# 在使用者家目錄底下的 public_html 為個人網頁目錄
    UserDir public_html
# 關閉 root 不能使用個人網頁
    UserDir disabled root
# 以下是預設的個人網頁目錄設定，若讀者要修改成不是放在 public_html 的話，# 要同時修改 UserDir
和 以下的 Directory 裡面的目錄位置。
    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    </Directory>
</IfModule>
```

設定好了以後，一般使用者可以在自己家目錄底下建立 public_html 的目錄來存放網頁，並且瀏覽器可以透過 `http://192.168.0.183/~UserName/` 來連到該使用者的個人網頁，其中 `UserName` 是使用者的帳號。

14.4 PHP 設定

14.4.1 設定檔設定

配合 Apache2 使用的 PHP 模組基本上不需要再修改，但是有些網頁是用 ASP style tag，或者有些舊的程式在傳遞變數的時候採用全域變數的做法，但在 PHP5 預設已經關閉全域變數傳遞來避免一些安全性問題。這時候若想要跑這些早期的程式，就得要自己修改一下 PHP 設定檔。所以這裡不能免俗的要稍微介紹一下

設定檔內容。提醒讀者一下，在 PHP 的設定檔是使用分號；來當成註解的開頭，讀者不要搞錯了唷！

PHP 設定檔重要部份擷選 /etc/php5/apache2/php.ini :

```
; 啓用 ASP style tag，通常用在早期使用 FrontPage 開發 ASP 程式時，
; 用 <? phpinfo(); ?> 這種 tag 來取代 <? phpinfo();?> 這種正規的 tag。
asp_tags = Off

; 每個 PHP 程式最大可以執行的時間 (秒)，通常不需要改動。
max_execution_time = 30

; 每次 PHP 程式分析要求的數據所能使用的最大時間 (秒)。
max_input_time = 60

; PHP 程式最大能使用的記憶體空間。如果有些程式需要比較多的記憶體，
; 可以自己到 32M 或者更高。
memory_limit = 8M

; 啓動全域變數，這是很不安全的唷！但是若你早期的程式在 POST 和 GET 的
; 變數沒有考慮到這個問題，就得要把它 On 起來。不過誠懇的建議你找本 PHP 的
; 書來弄懂這個問題，然後修正程式碼。筆者曾經處理過利用這種 bug 被入侵的站台，
; 可見打開這個選項的安全性危害很大，所以建議不要打開。
register_globals = Off

; 最大可以上傳的檔案大小。
upload_max_filesize = 2M
```

記得修改 php.ini 後，一定要重新啟動 Apache2 Server 才會生效唷！

14.4.2 使用 eAccelerator 來加速 PHP 程式

雖然 PHP 程式碼很容易撰寫，成為中小型網站最佳的解決方案，但是它不是沒有缺點的。由於 PHP 屬於直譯式語言，所以在執行的時候需要直譯器 (interpreter) 把程式碼即時翻譯成為機械語言 (opcode)，所以效能沒有那些需要編譯的 C++ 或者 Java 那麼快！由於是即時翻譯成 opcode，所以在 PHP 引擎上也很難最佳化一些迴圈策略等，這也是 PHP 效能不彰的另外一個原因。

所以市面上就出現了各種加速器，可以把 PHP 直譯後的 opcode 儲存起來，不需要每一次執行都需要在重新翻譯一次，此時執行 PHP 的效能會比沒有使用還要快，也就是可以透過加速器來減少 PHP 在伺服器上所使用的 CPU 資源。效能增加的多寡和網頁的複雜度有關，網頁越複雜可以最佳化的地方越多，也可以省下越多即時翻譯的時間，按照筆者的經驗是可以增加大約 10% 到 400% 左右的網頁伺服器效能，也就是不考慮頻寬問題的話，在相同的硬體資源可以平均增加大約一倍的使用者。

最有名的商業加速器就是 Zend Optimizer。而 PHP Engine 也是由同一公司維護的。該公司在 PHP 引擎是採取開放原始碼模式，讓大家都可以使用 PHP 來寫網頁，並且從服務與加速器來獲得公司的利潤。當然該公司也是 PHP 的核心維護者，所以相容性和效能是最好的，但是價格也是最貴的，這裡筆者要和讀者分享的是開放原始碼方案。

幾年前效能最好的 PHP 加速器是 Turck MMCache，但是該加速器的作者被 Zend 挖角，所以 MMCache 就幾乎就沒有發展了。後來一群人接下了 MMCache 的原始碼，並且重新改寫，成立了接下來要介紹的新專案 eAccelerator。

由於 eAccelerator 目前還沒有被 Ubuntu 收錄，而在 Ubuntu 套件庫裡面的 MMCache 不支援 PHP 5，所以以下示範我們將從 tarball 安裝。

首先從<http://www.eaccelerator.net/> 下載原始碼，由於 eAccelerator 要 0.9.5 以上才支援 Ubuntu 內附的 PHP 5.1，所以這裡筆者使用 0.9.5-rc1 作為範例，讀者可以安裝最新的版本。

安裝 eAccelerator :

```
# 下載原始碼，路徑可能會更動，讀者自己去下載抓吧！
dbtsai@www:~$ wget
http://bart.eaccelerator.net/source/0.9.5/eaccelerator-0.9.5-rc1.tar.bz2

# 解壓縮原始碼
dbtsai@www:~$ tar xvf eaccelerator-0.9.5-rc1.tar.bz2

# 安裝編譯時所需的套件和函式庫
dbtsai@www:~$ sudo apt-get install build-essential php5-dev
```

```
dbtsai@www:~$ cd eaccelerator-0.9.5-rc1

# 產生編譯所需的組態設定檔
dbtsai@www:~/eaccelerator-0.9.5-rc1$ phpize
dbtsai@www:~/eaccelerator-0.9.5-rc1$ ./configure

# 開始編譯
dbtsai@www:~/eaccelerator-0.9.5-rc1$ make
# 安裝到系統上
dbtsai@www:~/eaccelerator-0.9.5-rc1$ sudo make install

# 建立 eaccelerator 所使用的暫存目錄
dbtsai@www:~$ sudo mkdir /tmp/eaccelerator
dbtsai@www:~$ sudo chmod 0777 /tmp/eaccelerator
```

接下來要編輯 `php.ini` 把 `eAccelerator` 的模組啟動。讀者只需要在 `php.ini` 的最後面加上以下幾行就可以啦！

 在 `/etc/php5/apache2/php.ini` 最後面加上以下幾行：

```
; ... (略) ...
; 設定檔解釋請參考 http://eaccelerator.net/wiki/Settings
extension="eaccelerator.so"
; shm_size 為 eAccelerator 可以使用的記憶體大小，預設的 16 有點不夠，可以
; 改到 64 或 128，視你伺服器上的記憶體而定
eaccelerator.shm_size="16"
eaccelerator.cache_dir="/tmp/eaccelerator"
eaccelerator.enable="1"
eaccelerator.optimizer="1"
eaccelerator.check_mtime="1"
eaccelerator.debug="0"
eaccelerator.filter=""
eaccelerator.shm_max="0"
eaccelerator.shm_ttl="0"
eaccelerator.shm_prune_period="0"
eaccelerator.shm_only="0"
eaccelerator.compress="1"
eaccelerator.compress_level="9"
```


✎ 重新啟動 Apache2 讓剛剛的設定生效：

```
dbtsai@www:~$ sudo /etc/init.d/apache2 restart
```

這樣子就安裝好了唷！接下來可以測試到底效能增加多少，在業界網頁伺服器效能是使用 Apache Benchmark (ab2) 這個程式來測試。由於 ab2 會消耗掉待測主機的資源，並且會讓整個網路都趨於繁忙的狀態，所以你若測試你的伺服器，請在本機上或者同一個區域網路內測試，這樣才會有正確的結果。也因為這個程式會消耗掉幾乎所有被測試主機的資源，所以若對非你的主機測試的話，可能會被視為阻斷式 (DoS) 攻擊，所以別亂對非經過授權的主機測試喔！

✎ ab2 使用方式：

```
# 這個範例就是對該網址發出 5000 個請求，並且使用 20 的 thread
ab2 -n 5000 -c 20 http://xxx.xxx.xxx/index.php
```

在筆者自己的測試結果中，原先伺服器每秒可以提供 11 個動態請求，安裝了加速器後可以提供 32 個動態請求，大約增加三倍的網頁伺服器效能，也就是在同樣多的使用者下，可以約省下三分之二的 CPU 資源。

另外讀者也可以透過 14-2-2 裡面提及的 phpinfo() 函式來瞭解到 eAccelerator 已經被正確的安裝，此時會如圖 14-4-1 出現 eAccelerator v0.9.5-rc1 的字串。

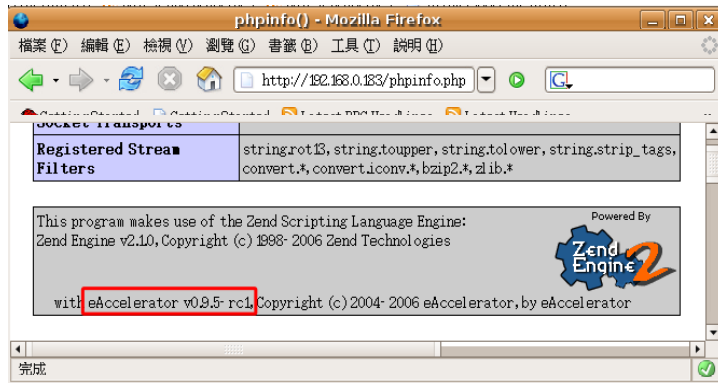


圖 14-4-1

配合之前提到的網頁文字檔壓縮來節省頻寬，在加上使用 eAccelerator 來節省處理動態網頁所需要的 CPU 資源，在筆者的個人網站可以讓最高線上人數在最多上三倍，這也是在有限的資源內壓榨出最好的效能。包含全球最大的開放原始碼

網站<http://sourceforge.net/> 都使用 eAccelerator 了，其它的網站更是不計其數。並且對於討論區或者一些大型的架站機如 xoops 等，加速效果更是明顯，聰明的讀者為何不給它一個機會去嘗試看看呢？

14.5 MySQL 設定

從 1995 年 MySQL 釋出到網路上給大眾使用，到現在已經過了十幾個年頭。隨著使用的人愈來愈多，越來越成熟，它也開始跨足原來大型商業資料庫市場。而原來的開發者組成了 MySQL AB 公司，以服務或者商業授權來獲得公司的利益。該公司讓 MySQL 擁有雙重授權，一個是 GPL 授權，一個是私有軟體授權，讓一般符合 GPL 法規的應用可以直接在網路上下載使用，若遇到問題，可以考慮付費給 MySQL AB 公司解決。若是要整合 MySQL 來作非自由軟體使用，那麼 MySQL AB 也有私有軟體的授權可以賣給你。

因為商業化的關係，MySQL 的開發者在有金費的開發環境下可以無後顧之憂的改善 MySQL，目前它漸漸擁有一些商業資料庫所需要的特性，因此傳統的資料庫軟體公司開始受到威脅，慢慢的也開放他們的產品在某些條件下免費使用，形成良性競爭。

由於 SQL 是一門獨立的學問，在這裡筆者無法詳細的討論，基本上 MySQL 的東西需要一本專書才能通盤瞭解，因此這裡只能給讀者一些很基礎的概念，例如怎樣備份和修改密碼等等。在管理資料庫上，若要使用文字模式來管理其實很不方便，讀者可以考慮安裝 phpMyAdmin 來管理，詳細使用方式請參考網路上文章或者相關書籍。

14.5.1 忘了或修改 MySQL 密碼

在 Ubuntu 或 Debian 的 MySQL 都有一個很特殊的帳號 debian-sys-maint，這個帳號是安裝完 MySQL 後自動產生的，並且擁有所有權限。它的帳號和密碼寫在 `/etc/mysql/debian.cnf` 裡面。為了安全因素，它的權限是 600，也就是只有 root 可以看的到它！由於這個帳號的密碼是隨機產生的，所以並不適合用在網頁中或者其他正式用途上，讀者應該把這個帳號當成救急用或者管理使用。還有一個需要讀者注意的是，讀者不能直接改 `debian.cnf` 來改這個帳號的密碼，這是沒有效

的！實際上它的密碼是放在資料庫，所以你改了該檔案後還是不會同步回資料庫裡面，Ubuntu 把密碼寫在那個檔案純粹是提醒讀者和救急使用的唷！若真的逼不得已要改，必須修改 `debian.cnf` 後，再用後面講到的方式修改該帳號的密碼到資料庫。

`debian.cnf` :

```
dbtsai@ubuntu:~ $ sudo more /etc/mysql/debian.cnf
# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host      = localhost
user      = debian-sys-maint          # 救急用帳號
password  = bsP8stUuZRqKvhre       # 救急用密碼
socket    = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
host      = localhost
user      = debian-sys-maint
password  = bsP8stUuZRqKvhre
socket    = /var/run/mysqld/mysqld.sock
```

當知道某帳號的密碼時，可以使用以下方式修改，這個方式一般個人 MySQL 使用者也可以使用。

在知道原始密碼的情況下修改 MySQL 密碼：

```
mysqladmin -u <帳號> -p password <輸入你要的新密碼>
# 按 ENTER 後輸入原來的舊密碼
```

當你不知道原先的密碼時，或者密碼遺失等等，可以使用救急帳號的帳號密碼來修改其他帳號的密碼。當你修改過 `debian.cnf` 要把他同步回資料庫，也可以使用這種方式。

用急救帳號或者 root 帳號修改其他帳號的 MySQL 密碼：

```
# 若使用者知道 MySQL root 密碼時，也可以使用同樣方式來修改其他帳號密碼。
# 當使用救急帳號時，其實可以不用輸入帳號和密碼，直接用以下指令引入 debian.cnf
# mysql --defaults-extra-file=/etc/mysql/debian.cnf


dbtsai@ubuntu:~ $ mysql -u <救急帳號> -p
```

```

Enter password: <---- 這裡輸入在 Debian.cnf 看到的密碼
mysql> use mysql
# 以下範例修改 root 密碼為 abcdef，同樣的方式可以改其他帳號 # 也可以用 set password for
root@localhost = password('abcdef');
# 不過這個是明確指定修改在主機 localhost 上的帳號。
mysql> update user set password=password('abcdef') where user='root';
# 將權限紀錄更新
mysql> flush privileges;
mysql> quit

```

最後一種情形是 debian.cnf 裡面的帳號莫名其妙的不能用，而且 root 帳號的密碼也搞丟了，這時候就比較麻煩了！此時需加上省略帳號認證的參數來略過帳號檢查並進入 MySQL 系統中。

 當什麼帳號密碼都不知道時，修改 MySQL 密碼的方式：


```

dbtsai@ubuntu:~ $ sudo /etc/init.d/mysql stop # 關閉 MySQL
dbtsai@ubuntu:~ $ sudo mysqld_safe --skip-grant-table &
dbtsai@ubuntu:~ $ mysql
# 接下來就和上一個範例一樣，輸入 SQL 指令來修改密碼，唯一不同的是
# 這個方式進入 MySQL 是不用密碼的！注意，改完後要趕快關閉 MySQL
# 再用正常的方式重新啟動，不然很容易被入侵！
dbtsai@ubuntu:~ $ mysqladmin -u root -p shutdown # 關閉 MySQL 伺服器
Enter password: <---- 輸入剛才改完的密碼
dbtsai@ubuntu:~ $ sudo /etc/init.d/mysql start # 啟動 MySQL 伺服器

```

14.5.2 資料庫備份

除了透過 phpMyAdmin 來用網頁介面備份資料庫，常見的作法是備份整個 /var/lib/mysql，還原的方式也很簡單，只要拷回去即可，通常會配合 crontab 定時把該目錄 tar 起來。

 使用 tar 備份完整資料庫：

```

sudo tar zcvf sql_backup.tgz /var/lib/mysql

```

但是透過備份 `/var/lib/mysql` 的交換性比較差，有時候換到不同版本的 MySQL 主機可能會不相容，所以這裡筆者比較常把 SQL 資料 dump 到一個文字檔裡面，這裡面還會包含還原資料庫的 SQL 指令，所以很容易帶到不同主機上還原使用。

使用 mysqldump 備份資料庫：

```
# 使用以下指令可以用 root 身份備份 mysql 這個資料庫到 filename.sql
# 若要寫在 crontab 裡面使用，那麼輸入 MySQL root 帳號的密碼是很麻煩的！
# 此時筆者會配合 --defaults-extra-file=/etc/mysql/debian.cnf
# 自動輸入帳號密碼。若要備份整個資料庫的話，需要用 --all-databases 的參數。
mysqldump -u root -p --databases mysql > filename.sql
```

以下範例將會引入 `debian.cnf` 來達成不需要輸入密碼的備份，且將系統上所有的資料庫都備份起來，同時也壓縮成 `gz` 檔。因為不需要輸入密碼，很適合放在 `crontab` 裡面自動定時備份。`dump` 下來的 SQL 原始檔是文字檔，有時候在傳遞時會因為不同系統換行等問題把檔案損毀，所以壓縮起來可以避免這個問題，且大約會讓備份的檔案減少到原來的 30% 左右。

使用 mysqldump 的範例：


```
# 以下有兩行，第一行最後面的 \ 代表換行。如果不打 \ 也可以，就接下去打，
# 不用換行。讀者也可以用 bzip 取代 gzip，壓縮率會比較好，不過壓縮
# 時間會多很多倍！

dbtsai@ubuntu:~ $ sudo mysqldump --defaults-extra-file=/etc/mysql/debian.cnf \
--all-databases | gzip > backup.sql.gz

# 查看備份檔內容
dbtsai@ubuntu:~ $ zcat backup.sql.gz
-- MySQL dump 10.10
-- Host: localhost      Database:
-- Server version      5.0.22-Debian_0ubuntu6.06.2-log ... (略) ... dbtsai@ubuntu:~
$
```

上面使用 `mysqldump` 下來的資料庫備份檔本質是文字檔，那我們要如何復原到資料庫裡面？其實我們只要使用 `mysql -u root -p < backup.sql` 來匯入該文字檔。遇到壓縮過的資料庫檔案，其實道理是一樣的。因為之前舉的例子是壓縮過

的，所以在以下範例中筆者使用 `zcat` 透過 `stdin` 送到 `mysql` 這個程式，並且利用 `debian.cnf` 裡面的帳號密碼來達成不需要輸入密碼的功能。

 將壓縮後的 `backup.sql.gz` 資料庫備份還原範例：

```
dbtsai@ubuntu:~ $ zcat backup.sql.gz | sudo mysql \  
--defaults-extra-file=/etc/mysql/debian.cnf
```

14.5.3 開啟 port 3306 讓非本地端主機連線

為了安全性因素，預設 MySQL 是不允許 client 端從非本機端連線查詢資料。在預設的行為中，client 端只能在本機透過 `/var/run/mysqld/mysqld.sock` 來和 MySQL 伺服器溝通，也就是在傳遞資料時是不透過 port 3306 而是透過該 socket 來提供內部存取和查詢。

若讀者系統的規劃是 MySQL 主機獨立出來，此時網頁主機就需要透過 port 3306 來和 MySQL 主機連接，這時候就必須在 MySQL 上開啟 port 3306。或者讀者想要讓該 MySQL 主機提供給很多非本地端的使用者使用，也需要打開 port 3306。在打開 port 3306 後，建議讀者要設防火牆來限制可以存取該 MySQL 伺服器的 client 端位置，不然很容易被入侵。

要讓非本地端使用者可以使用 MySQL 只要如以下範例去修改 `/etc/mysql/my.cnf`，整個過程只要註解掉一行，接下來重新啟動 MySQL 伺服器讓新設定生效。

```
...(略) ...  
# 註解掉下一行來啟用非本地端存取  
# bind-address          = 127.0.0.1  
...(略) ...
```

CHAPTER 15

DHCP 伺服器

DHCP (Dynamic Host Configuration Protocol) 是要讓使用者端自動取得網路組態設定的協定。當沒有使用這個協定時，網管需要每台電腦都給與網路 IP，光是規劃就很麻煩了，而且大部份的使用者連怎樣設定 IP 都不大清楚，最後還得要網管親自出馬。

因此透過 DHCP Server 來讓各個 DHCP Client 自動設定好網路是有效率的事情，網管從此之後就不需要一台一台的設定網路了，它可以自動配發某個網段的 IP，也可以用 MAC Address 來指定某幾台電腦使用固定 IP。在公司或學校，以及家中使用 IP 分享器來自動取得 IP，都是使用 DHCP 這樣的技巧。

在架設前，你應該先確認你所屬的區網能不能讓你架設 DHCP Server。當原來區網上已經有 DHCP Server，若你再架上第二台，會導致 Client 端不知道要聽那台配發的 IP，會造成網路大亂喔！

15.1 DHCP 的運作理論

當 DHCP Client 要和 DHCP Server 取得 IP 時，DHCP Client 還沒有設定好網路，那怎樣和 DHCP Server 要 IP 呢？記不記得在「網路概論」的地方談過區域網路內可以透過廣播的方式，把封包丟到所有區網上的電腦呢？DHCP Client 就是利用這種方式來連接 DHCP Server 的。簡單的流程如下：

1. 首先 DHCP Client 丟出一個 UDP 封包（DHCP Discover），裡面來源 IP 和 Port 分別填 0.0.0.0、68，目的地 IP 和 Port 分別填 255.255.255.255、67。然後把它放在網路層包裝，並在框架表頭填上自己的 MAC Address 後，送出去廣播。
2. 此時整個區域網路的電腦都會收到這個封包，但是解開以後，會發現目的地是 255.255.255.255，所以一般的電腦都會把這個封包丟棄。但是當 DHCP Server 收到後，會發現這是要給它的，於是它會把該封包撿起來。之後會比對來源的 MAC Address 是否在 DHCP Server 上有設定要給與固定 IP，若有的話，則依照 Server 上的設定值傳回設定值給 Client 端。若沒有的話，會選取可發放 IP 裡面還沒有人用的 IP 給 Client 端，這時候傳回 Client 端的封包稱為 DHCP Offer。注意唷，在區網內只要有 MAC Address 就可以傳送封包，不需要等到設定好 IP 組態。
3. 當 Client 收到 Server 給它的設定後，會先用 ARP 檢查該 IP 是否已經有人使用，若已經有人使用，Client 就會發一個 DHCP Decline 的封包給 Server，告訴 Server 不會使用該 IP，再重新從步驟 1 開始要其他的 IP。若 Client 端測試為該 IP 沒有人用，會發一個 DHCP Request 給 Server 確認使用該網路設定。
4. 最後 Server 收到 DHCP Request 後，Server 端會再發一個 DHCP Pack 封包，確認給與的 IP 租約生效，此時才完成整個 DHCP 協定的流程。
5. 在步驟 2 的時候，Server 會給 Client 端一個合約時間，當使用超過了合約時間就需要重新續約，也就是再重新跑一次 DHCP 的流程。通常當 Server 端可以分配的 IP 很充足時，續約會優先給同樣的 IP，若當 Server 端的 IP 用完後，可能會無法續約，就需要排隊等候了。

15.2 DHCP 實際安裝

表 15-1 DHCP Server 重要的相關檔案

功能	設定檔
控制 DHCP 啟動或關閉的 script 檔	/etc/init.d/dhcp3-server
DHCP 設定檔	/etc/dhcp3/dhcpd.conf
DHCP Client 租約紀錄檔	/var/lib/dhcp3/dhcpd.leases

要安裝 DHCP Server 很簡單，只要輸入以下指令就好了。

安裝 DHCP Server :

```
sudo apt-get install dhcp3-server
```

安裝完後，它會自動把 /etc/init.d/dhcp3-server soft link 到 rc2.d ~ rc5.d，也就是開機後在這些 runlevel 中都會自動啟動 DHCP 伺服器。並且會在安裝完後順便啟動，當然此時設定檔還沒有設好，所以啟動會失敗。因此設定好設定檔後，讀者需要再自己重新啟動。

重新啟動 DHCP Server :

```
sudo /etc/init.d/dhcp3-server restart
```

啟動 DHCP Server :

```
sudo /etc/init.d/dhcp3-server start
```

關閉 DHCP Server :

```
sudo /etc/init.d/dhcp3-server stop
```

上面的指令也可以用 invoke-rc.d 取代，不過筆者還是習慣執行 init.d 裡面的控制 script。

重新啟動 DHCP Server :

```
sudo invoke-rc.d dhcp3-server restart # 後面參數可以是 stop, start
```

若要讓它開機時不要自動執行，只要刪除 rcX.d 裡面的 soft link 即可。這裡提醒讀者所採用的是 update-rc.d 的方式。

 設定為開機時不要自動啟動 DHCP Server :

```
sudo update-rc.d -f dhcp3-server remove
```

設定為開機時自動啟動 DHCP Server :


```
sudo update-rc.d -f dhcp3-server defaults
```

15.2.1 發放浮動式 IP

因為 Ubuntu 預設的設定檔很長，所以這裡我們把它備份後，自己從頭來寫設定檔會比較方便。

 備份原來的 DHCP Server 設定檔 :

```
sudo mv /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf.bak
```

 開始寫新的 DHCP Server 設定檔 :

```
sudo gedit /etc/dhcp3/dhcpd.conf
```

 設定檔 dhcpd.conf 範例 :

```
# 這個設定是當用戶續約使用 IP 時，是否要重新確認新的 DNS Server
# 給客戶使用，由於 DHCP v2 不支援，所以為了相容性原因建議設為 none !
ddns-update-style none;

# 預設的租約秒數（這裡可以不要設，也可以直接設定在後面的區域變數裡）
default-lease-time 600;

# 最大的租約秒數（這裡可以不要設，也可以直接設定在後面的區域變數裡）
max-lease-time 7200;

# 設定網域 domain，當客戶端收到後，會加到 resolv.conf 裡面
# 而實際功能請讀者參考網路設定那一節，因此這個設定是可有可無的！
# 也可以這裡不寫，寫在區域變數裡。
option domain-name "emook.org";
```

```
# 告訴客戶端可以使用的 DNS Server，客戶端收到後，
# 可以自動設好 DNS，這個設定也可以寫在區域變數
option domain-name-servers 168.95.192.1, 168.95.1.1;

# 和 dhcp log 檔有關，讀者只需要寫上去即可
log-facility local7;

# 設定該網段的識別代號 (NetID)，值為該網段第一個 IP
# netmask 就依照網段大小設定即可
subnet 192.168.0.0 netmask 255.255.255.0 {

    # 設配發的 IP 範圍，可以有數個
    # 在這個例子，將配發 10 到 20 和 35 到 45 的 IP
    range 192.168.0.10 192.168.0.20;
    range 192.168.0.35 192.168.0.45;

    # 設定所使用的 gateway (router)
    option routers 192.168.0.1;

    # 廣播地址 (broadcast)，這可以從 netmask 和 subnet
    # 網段算出，所以可有可無，此值為該網段最後一個 IP
    option broadcast-address 192.168.0.255;

    ### 若全域變數有設的話，設到這裡就可以工作了
    default-lease-time 600;
    max-lease-time 7200;
    option domain-name "emook.org";
    option domain-name-servers 168.95.192.1, 168.95.1.1;
    ### 當全域變數和區域變數都有設定時，以這裡的區域
    ### 變數優先使用
}
```

由於 DHCP Server 可能插很多塊網路卡，服務數個網段，所以有些設定值可以寫在區域變數內，不用做全域設定。Server 上的網卡應該要有 DHCP Server 所服務的網段 IP，當 DHCP Server 啟動錯誤時，讀者可以檢查看看，在 ifconfig 網路卡裡面的 IP 設定是否有在該服務網段內，匹配不起來是不能啟動的。


所謂的預設租約時間是當 Client 在和 Server 要 IP 時，若 Client 沒有提出要求，就用這個時間當租約時間。當 Client 有提出比較長的時間要求，最大不能超過最大租約時間。

讀者設定好後，記得重新啟動 DHCP Server 服務，讀者可以透過以下指令看到 DHCP Server 真的在運作呢！

 看看 DHCP Server 是否真的在服務：

```
dbtsai@ubuntu:~ $ sudo netstat -upln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address   Foreign Address PID/Program name
udp        0      0 0.0.0.0:67      0.0.0.0:*       21861/dhcpd3
```

讀者也可以透過租約紀錄檔，瞭解到有哪一些機器和 DHCP Server 要 IP。

 租約紀錄檔：

```
dbtsai@ubuntu:~ $ more /var/lib/dhcp3/dhcpd.leases

lease 192.168.0.44 {
  starts 6 2006/08/19 14:36:59;
  ends 6 2006/08/19 14:46:59;
  tstp 6 2006/08/19 14:46:59;
  binding state free;
  hardware ethernet 00:11:24:29:3c:bd;
}
```

15.2.1 發放固定式 IP

若要對單一主機設定固定 IP 的話，其實也不難，我們先編輯設定檔吧！

 編輯 DHCP Server 設定檔：

```
sudo gedit /etc/dhcp3/dhcpd.conf
```

設定發放固定式 IP 比浮動式 IP 只有多幾個設定而已，相信對讀者不會太難。

設定檔 dhcpd.conf 範例：

```

ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
option domain-name-servers 168.95.192.1, 168.95.1.1;
log-facility local7;

subnet 192.168.0.0 netmask 255.255.255.0 {
    ..(省略浮動 IP 設定) ...
    option routers 192.168.0.1;
}

# 設定 ubuntu 這台機器取得固定 IP，ubuntu 是可以自己取的名稱
host ubuntu{
    # 設定該機器的網卡 MAC Address
    hardware ethernet 00:11:22:33:44:55;

    # 設定要給該機器的 IP Address
    fixed-address 192.168.0.150;
}

```

如以上範例，當有一台 client 的 MAC Address 是 00:11:22:33:44:55 並且向 DHCP Server 取得 IP 時，Server 就會自動給它 192.168.0.150 這個固定 IP。讀者或許會覺得奇怪，裡面沒有 netmask，gateway 等設定，網路怎麼會通？其實在這些固定配發 IP 的設定前一定要先有 subnet 來宣告該網段 netmask、gateway 和 NetID 等組態。

所以當你設定好配發固定 IP 後，若有錯誤，要檢查一下你配發的 IP 是否已經在 subnet 裡面有先宣告該網段。若只要配發固定 IP，不要浮動 IP，那麼 subnet 也要宣告，但裡面的 range 設定可以省略。

設定好後，重新啟動服務，就應該可以正常運作了唷！

CHAPTER 16

Samba Server 與 Windows 互相分享檔案

在 Linux 下我們可以透過 Samba，來跟 Windows 分享檔案，也就是說 Samba 可以讓你在使用 Unix Like 的作業系統時，用 Samba 來分享檔案。Samba 其實是 SMB 的縮寫（Server Message Block），會命名為 SAMBA 則是因為，當初 SMB 的程式開發者為了註冊商標的原因。而 Samba 運用的是 NetBIOS over TCP/IP 的技術，這個技術最早是 IBM 發展出來的 NetBIOS 的改良版本，那為什麼會談到這個呢？因為在 Windows 下設定網路時，常常會看到 NetBEUI（NetBEUI 是 NetBIOS Extended User Interface 的縮寫，這也是 NetBIOS 後來的發展改良版本，不過技術上是不太一樣的），而微軟的網路架構裡是採用 NetBEUI，所以 Samba 可以透過 NetBIOS，在 NetBIOS 架構上的網路跟 Windows 的機器溝通。

檔案分享的型態可以分成兩種類型，第一種是分享 Linux 裡的檔案給其他的電腦，這個是 Samba Server 的部份。第二種是跟網路上的芳鄰裡的 Win32 電腦，要求它所分享的檔案。在這個章節筆者將介紹 Ubuntu 裡的**共享資料夾**，透過共享資料夾的圖型介面將 Samba Server 架設起來。接著再解說 Samba 的設定檔，讓讀者可以輕鬆的學會 Samba 的應用。

關於更詳細的 Samba 命名原由，與 Samba 的運作原理，以及更多的應用，可以參考鳥哥大長輩所寫的文章。鳥哥的 Linux 私房菜 SAMBA server 篇^❶。

16.1 用圖型介面來裝上 Samba server

讀者可以開啟系統-->管理-->共享資料夾，來安裝 NFS、SMB，如下圖：



圖 16.1 安裝 SMB、NFS

^❶ http://linux.vbird.org/linux_server/0370samba.php

必需輸入系統管理者的密碼以繼續安裝



圖 16.2 輸入系統管理者的密碼

系統正在自動下載套件

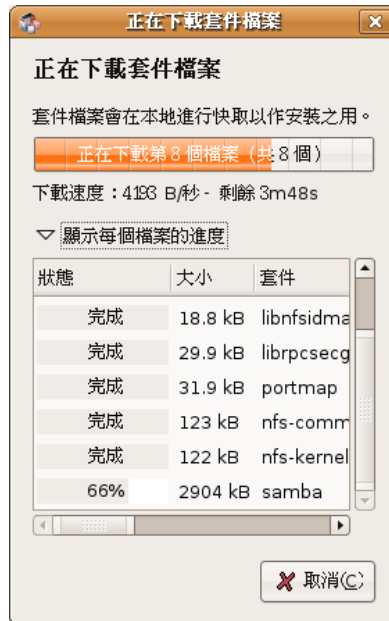


圖 16.3 下載套件中

系統正在安裝 SMB、NFS



圖 16.4 安裝套件中

安裝完成直接按下關閉即可，接著會出現視窗問你，要分享哪個資料夾、以及用什麼方式分享檔案。這裡我們選 SMB，然後也選擇一下要分享的資料夾。



圖 16.5 分享的設定

設定好了以後會出現下面這個共享資料夾的視窗，讀者也可以另外再增加別的目錄哦！

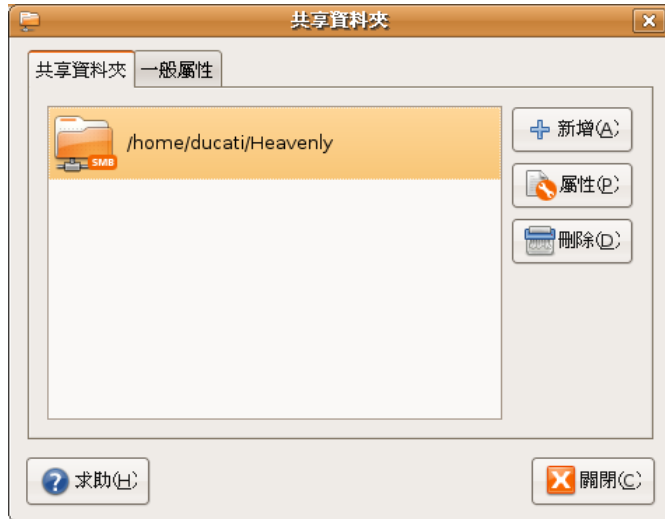


圖 16.6 你所分享的資料夾

接下來重要的是設定你的 SMB 屬性，第一項是設定最重要的 Workgroup（工作群組），第二項則是 WIN 伺服器。筆者在這裡要解釋一下，Workgroup 要一樣才可以在同一個群組裡分享檔案。WIN 伺服器就要看你的 Ubuntu SMB，在這個 Workgroup 是否要當一個 Server，一個 Workgroup 一定要有一個 Server，看

你是要用 Ubuntu 來當 Server，還是要用 Windows 來當 Server，這就要看你 Workgroup 的性質了。用 Linux 來當 Server 有很多好處，第一點就是穩定，並且比較不用擔心糟到病毒的傷害，而且又可以 24 小時不關機的運作。那麼要是一個 Workgroup 裡已經有台電腦是 Server 的話，而你只是要下載別台機器的檔案，WIN 伺服器的選項可以不用打勾。



圖 16.7 屬性設定

那麼這樣子 Samba Server 最基本的設定就弄好囉。

16.2 設定使用者帳號密碼

別急別急！雖然在上面的小節我們設定好了 Samba，不過還要為使用者的帳號加入到 Samba 哦！加入前要先注意一下，我們要加入到 Samba 裡的帳號，在系統上是不是也有了。例如我們在 SMB Server 裡有個帳號叫 ducati，那麼讀者可以用這個指令來加入 ducati 這個帳號。

```
ducati@ubuntu:~$ sudo smbpasswd -a ducati
Password:
New SMB password:
Retype new SMB password:
```

先輸入系統管理者的密碼，接著在輸入 ducati 在 Samba 的密碼，Samba 的密碼可以跟 ducati 這個 user 原本的密碼不一樣，因為 Samba 的帳號跟系統上的帳號是分開管理的。設定不一樣的密碼同時也可以增加安全性。這個指令很重要哦！那麼我們來看看這個指令還有什麼參數可以用。

```
ducati@ubuntu:~$ smbpasswd -h
When run by root:
    smbpasswd [options] [username]
otherwise:
    smbpasswd [options]

options:
  -L                local mode (must be first option)
  -h                print this usage message
  -s                use stdin for password prompt
  -c smb.conf file  Use the given path to the smb.conf file
  -D LEVEL          debug level
  -r MACHINE        remote machine
  -U USER           remote username
extra options when run by root or in local mode:
  -a                add user
  -d                disable user
  -e                enable user
  -i                interdomain trust account
  -m                machine trust account
  -n                set no password
  -w PASSWORD       ldap admin password
  -x                delete user
  -R ORDER          name resolve order
```

接下來可以開啟一台 Windows 的電腦，並設定好網路上的芳鄰（注意：工作群組要一樣）。

用 Windows 登入 Samba Server

然後再開啟網路上的芳鄰，按一下右邊的檢視工作群組電腦，就可以看到 Ubuntu Samba Server 囉，然後輸入帳號密碼就可以了。

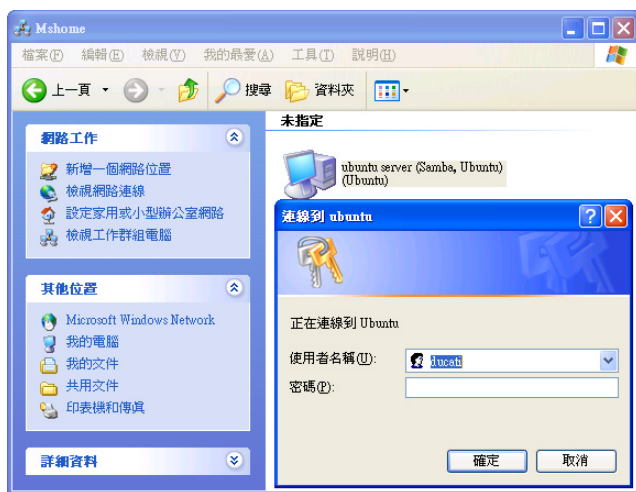


圖 16.8 用 Windows 登入 Samba

16.3 用 Ubuntu 進入 Windows 網芳鄰

要用 Ubuntu 來讀取網路上的芳鄰，真是一件很方便的事，甚至不用裝上 SMB 就可以了。只要開啟 位置-->網路伺服器，就可以直接看到網路上的芳鄰裡的內容了。

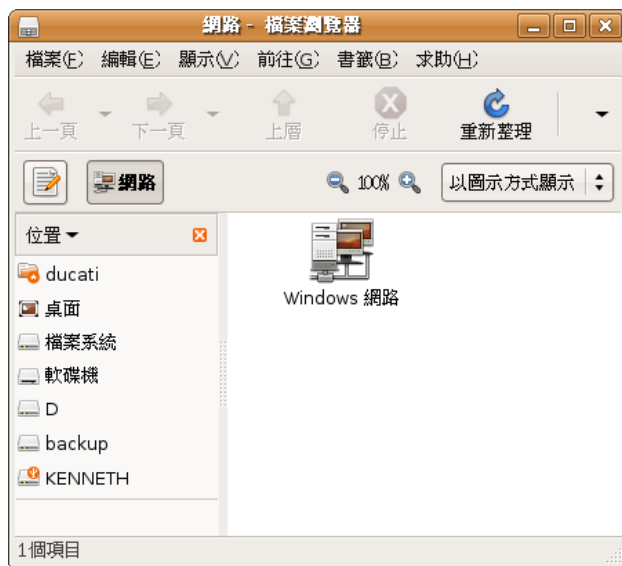


圖 16.9 網路上的芳鄰裡的內容

16.4 SAMBA 的相關設定

前面的小節都只是 Samba 很簡單的設定及使用方式，那麼如果讀者想要更深入的了解 Samba 還有什麼功用，就要接著看下去囉！首先讓我們來了解 Samba 的設定檔：

```
/etc/samba/smb.conf
```

Samba 所有的設定都可以在這個設定檔裡完成。接著來看一下這個設定檔裡的主要參數是做什麼用的。

```
##### Global Settings #####

[global]

    workgroup = MSHOME
    #這一行是設定工作群組

    server string = %h server (Samba, Ubuntu)
    #在網路上的芳鄰看到的主機名稱 (%h 是 host name 變數)

#### Debugging/Accounting ####

    log file = /var/log/samba/log.%m
    #可以指定 log 檔的位置

# Put a capping on the size of the log files (in Kb).
    max log size = 1000
    #log 檔的大小，以 Kb 為單位，0 代表沒有限制

##### Authentication #####

; security = user
#安全性等級：
# user 代表依照使用者的帳號，可以分配不一樣的檔案分享權限。
# share 代表不用密碼就可以分享任何檔案與資料夾，是屬於最不安全的等級。
# server 代表需透過其他的 smb server、win server 來驗證帳號密碼。
```

```
encrypt passwords = true
#密碼傳遞時是否加密

; guest account = nobody
# 這是用來對應 Windows 上沒有帳號的 user，讀者也可以弄個帳號來統一管理沒有帳號的 user，
# 若不想設定就設成 nobody 即可。

invalid users = root
# 無效的使用者帳號，每一台 Linux 上都有 root，除非是自己修改過，
# 在這裡將 root 填入是爲了防止有心人士的猜測。以提高安全性

; unix password sync = no
是否同步 samba 與系統上的帳號密碼，通常是 no。

hosts allow = 192.168.100
#限制可以登入的 ip，如上所示只要是在 192.168.100.x 這個網段的 ip 都可以登入。

client code page = 950
#設定語系，950 爲繁體中文

#===== Share Definitions =====

writable = no
#是否可以寫入

path = /home/ducati/heavenly
#分享的路徑

read only = yes
#是否唯讀

public = no
#是否公開分享
```

接著我們可以來看一下剛才所加入分享的目錄，以家目錄下的 heavenly 目錄做個例子講解：


```
[Heavenly]
path = /home/ducati/Heavenly
available = yes
browsable = yes
public = yes
writable = no
```

剛才加入分享的目錄可以在 `/etc/samba/smb.conf` 的最下面看到。所以看到路徑是 `/home/ducati/Heavenly`，而這個目錄可以瀏覽、有公開、但是不能寫入。那如果想要讓 user 有寫入(上傳檔案)的權限呢？其實可以在下面再加上 `create mode = 0666`，同樣的也要在終端機輸入 `chmod -R 666 /home/ducati/Heavenly`，這樣這個資料夾才有被寫入的權限。也可以加上 `write list = ducati`，這樣子就只有 ducati 可以上傳(寫入)的權限，而其他人不行但是可以看。那麼也可以設定只有某些使用者可以登入，用這一行 `valid user = ducati` 加在下面，如果要多增加 user，只要用逗號分隔即可，例如：`valid user = ducati, dbtsai`，不過這樣別的用户就連看都不能看了。以上的參數都可以視個別的目錄以及使用者分別設定。

16.5 Samba Server 的啟動與關閉

很簡單！Samba 的啟動檔如下：

```
/etc/init.d/samba
```

有這些參數可用：`force-reload | reload | restart | start | stop`

例如想要停止 Samba Server 的話，可以在終端機下輸入

```
$ sudo /etc/init.d/samba stop
```


CHAPTER 17

NFS 伺服器

NFS (Network File System) 是由昇陽 (Sun) 最先開發，它的目標是讓不同機器能夠跨平台共享相同的檔案資源。簡單來說，可以看成是一種遠端的檔案系統。也因為跨平台的目標，所以幾乎所有的 Unix Like 平台都支援 NFS，如 IBM AIX、Sun Solaris、HP Unix、GNU/Linux 和 FreeBSD 等，都可以透過 NFS 來共享檔案和目錄資源。目前在 MS Windows 上雖然有 NFS Client 可以安裝，但幾乎是商業軟體。

所以大家常在 Unix Like 系統中使用 NFS 來共享檔案資源。或許讀者會奇怪，為何不用 Samba 網芳分享，這樣 MS Windows 和 Unix 都可以互相連線？這是因為 NFS 效能比 Samba 好很多，所以常常在平行運算的超級電腦或者商業大型系統上，使用一台 NFS Server，其他 Server 透過光纖通道連線到 NFS 檔案伺服器取得資料。

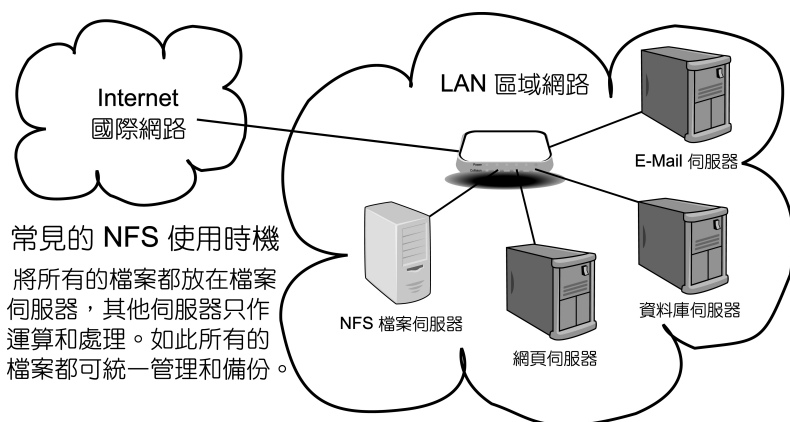


圖 17-1

如圖 17-1，實務上常常架一台 NFS Server，並且上面裝有磁帶機可以定時的備份檔案。其他的伺服器就把 NFS Server share 出來的目錄掛載起來，實際上所有網頁和資料等檔案都是放在 NFS Server 上，這些 Server 就只單純運算與回應客戶端請求。這樣的好處是我們只要維護好一份檔案系統即可，並且備份和資源分配也會比較容易。也因為 NFS Server 通常使用 Raid 磁碟陣列系統，所以整個架構可以達成高效能且高可靠性的服務。

17.1 NFS 運作理論

NFS 有好幾個版本，最新版是 NFSv4。而 NFSv1 到 NFSv4 都是透過 RPC 來運作，所謂的 RPC (Remote Procedure Call) 是遠端程序呼叫。當跑一個程式時，該程式呼叫另外一個程式去做事情然後把結果傳回來稱為 Procedure Call，所以當一個程式呼叫在不同主機的程式去處理事情並傳回結果，我們就稱為 Remote Procedure Call。對於呼叫別人的程式來說，它會感覺到被呼叫的程式也是在本機執行的。所以 RPC 有個好處就是在寫程式的時候，只要管好呼叫的部份，我們不必實際去處理網路封包等細節，這些細節就給 RPC 系統處理即可，這樣程式也會好寫很多。故 RPC 程式要處理遠端傳輸的問題，只要寫好 procedure call 來呼叫遠端主機的程式讀取或寫入硬碟即可，這大大簡化 coding 的難度。

NFS 的實作就是靠 RPC 了，所以 NFS 是不處理網路封包的問題，它只從 Client 呼叫 Server 上的 NFS procedure，並且告訴該 remote procedure 想要幹嘛，而

封包實際的傳輸就透過標準的 RPC 協定來達成。故在 NFS 系統上，Client 端和 Server 端都要安裝 RPC 系統，而 RPC 在 Linux 核心裡已經有內建。

🔗 察看系統上的 RPC 程序：

```
# 後面可以不加網址，查詢本機 RPC 程序
dbtsai@ubuntu:~ $ rpcinfo -p 192.1680.183

程式採用的協定連接埠

 100000      2    tcp    111    portmapper
 100000      2    udp    111    portmapper
 100003      2    udp    2049   nfs
 100003      4    tcp    2049   nfs
 100005      1    tcp    806    mountd
 100005      2    udp    803    mountd
 100024      1    udp    887    status
 100024      1    tcp    890    status

... (略) ...
```

現在我們來談談 Portmap Daemon 這個服務！通常當一個 RPC 程序啟動時，若需要使用網路傳遞資料，需要先有個 port 吧！所以 portmap 就是把 RPC 程序啟動後，所使用的 port 紀錄起來。因此當 Client 端要連到 RPC 程序前，會先連到 portmap 所使用的 port 111，向它查詢該 RPC 程序所使用的 port。

這時候就產生一個議題了，每次 RPC 程式向 portmap 註冊的 port 可能都不一樣，這樣造成我們很難寫防火牆規則去擋其他不開放的 port，這也是為什麼使用 RPC 時常會讓系統安全性不佳的原因。由於這種安全性議題 NFS 和 NIS 等使用 portmap 的 RPC 服務通常限制在區域網路內使用，甚至把所有使用 RPC 的機器都放到虛擬 IP 網段內，再透過 NAT 來對應外部實體 IP 的 port 到內部虛擬 IP 想開放的 port。

這個問題目前在 NFSv4 已經解決了，也就是不必再透過 portmap 來對應 port 和 RPC 程序，因為它改成使用固定的 port 2049 給 Client 連接，此時防火牆規則就很好寫，不需要考慮到 NFSv3 Server 使用的 port 是不固定。

NFSv4 提供的新功能包含了完整的 Unicode 支援，如同 Samba 的 CIFS 協定，會在檔案傳輸的時候採用 Unicode (UTF-8) 來傳輸檔名。舊版的 NFSv3 在傳輸檔名時，是用本地端的 locale 來傳輸檔名，也就是傳輸檔名原始資料。但這會有

一個問題，如果當 A 機器使用 Big5 Locale，B 機器使用 UTF-8 Locale，那麼我們就無法透過 NFSv3 來正確傳輸中文檔名，因為 NFSv3 只能傳輸原始 locale 所用的編碼，不提供轉碼阿！所以在從前兩台 Unix 主機必須要使用相同的 locale 才能正確的傳輸中文檔名。

但是在 NFSv4 的話，理論上情形就改觀了！當一台機器 Locale 是 Big5，另外一台是 UTF-8，在傳輸檔名時，都會先轉成 Unicode (UTF-8) 傳輸，到本地後再轉成該機器所使用的 Locale 編碼。因此可以解決不同 locale 之間中文亂碼的問題。不過轉碼的部份，因為筆者所有的機器都使用 UTF-8，筆者沒有機會去試，若讀者有嘗試成功的話，希望可以和筆者分享一下 ^_^。

NFSv4 還有其他的優點，例如增加了傳輸加密的功能 (Kerberos 5)，因此也更適合直接在 Internet 上跨不同網域架設 NFS Server 和 Client，不必擔心竊聽的安全問題。不過這在本書中沒有討論到，有興趣的讀者可以上網找資料或和筆者討論。

在 NFSv4 對其他 Unix 主機的支援性方面，IBM AIX、Sun Solaris 和 HP Unix 都已經很穩定的支援，因此 NFSv4 和其他 Unix 主機溝通目前都已經沒有問題。但是目前在 Linux 核心上 NFSv4 仍然被標示為 EXPERIMENTAL，也就是還在實驗階段。筆者自己使用的情況是在 Ubuntu dapper 2.6.15 核心沒有問題，但是在自己編的官方 2.6.17 核心 client 端會不穩定，再更新到 2.6.18 就沒有問題了，可見真的有一點實驗性質。

因為它有加密的功能和容易設定防火牆的特性，在筆者管的 Server 是同時使用 NFSv3 和 NFSv4。NFSv3 拿來區網內使用，NFSv4 拿來穿越防火牆給校外的 Server 加密連線。所以要使用 NFSv4 或 NFSv3 讀者要自己考量和測試！

17.2 NFS Server 主機規劃

設定其實是不難，難的是資源分享的規劃和權限設計。這個萬一沒弄好，別人有可能可以隨便對你的 NFS Server 存取資料。所以在設定前，先想想看要怎樣規劃檔案分享吧！讀者首先要想的是 NFS Server 可以提供給哪些 Client 機器使用？基本上它不提供帳號認證的方式，所以它使用 IP 範圍來控管 Client 端。再來你要考慮分享出去的目錄權限問題，權限分配有以下幾種方式。

1. `no_root_squash` : 這種權限設定就有點危險了！它的意思是若 Client 端帳號在 Server 上也有的話，那麼在 Client 端操作該分享目錄時，會有該帳號在 Server 上的權限。危險的地方是在當 Client 端是 root 時，那麼可以同時取得 Server 上在分享出來的目錄下完整權限。若是 Server 端沒有該帳號，但是 Client 端有該帳號時，就會變成用 nobody 來操作該分享目錄唷！我們通常使用 `root_squash` 會比較安全，且經常會配合 NIS 一起使用。
2. `root_squash` : 這個基本上和 `no_root_squash` 一樣。不同的是當 Client 端用 root 去操作時，就會變成匿名的 nobody 權限，所以就安全多了，Client 端也無法去做大規模的破壞。通常會配合 NIS 一起使用。
3. `all_squash` : 這代表了不管 Client 端用什麼帳號，在該分享目錄的權限都只有 nobody。在開放給所有人使用的匿名 NFS Server 通常會用這樣的設定。
4. `anonuid/anongid` : 當 Client mount 起來後，不管用什麼帳號，在該目錄的權限只有 `anonuid` 指定的帳號與 `anongid` 指定的群組。當你將 `anonuid` 和 `anongid` 都設成 nobody 的 id 時，其實和 `all_squash` 有一樣的意義！

考慮好權限問題後，讀者還要考慮該目錄是只能讀 (ro)，或者可以讀寫 (rw)。當要開放給大家只能抓東西，這時候都會配合 ro 的設定。若是配合 NIS 讓每一部機器的帳號和權限都一樣的話，那可以把家目錄都放在 NFS Server 上，並使用 rw 的設定，這時候不同機器可以負責不同的運算功能，例如網頁和 Mail Server 等，但都是存放在同一個檔案系統上。

接下來要考慮可以掛載該分享目錄的 IP 範圍，這有兩種表示法，一種是直接寫上它的 IP，筆者不建議用網址 (xxx.mook.org) 的方式，除非你是寫死在 hosts 裡面，不然有可能會因為 DNS Server 被 hack 導致你的 NFS Server 也被攻陷！另外一種方式是用網域的代表方式。例如你想要整個 192.168.1.xxx 的 C Class 網域都可以連線到該台 NFS Server，那你可以寫成 192.168.1.0/24。後面的 24 是 netmask 255.255.255.0 的簡寫，換算方式是把該 netmask 轉成二進位後，從前面數過來看有幾個 1。如每一個 255 都由 8 個 1bit 組成，所以總共 3 個 255 就是 24 了！計算結果是整個 C Class 簡寫為 24，B Class 簡寫為 16，以此類推。通常要分享到整個網域時，常會把主機全部放在防火牆裡面，這樣 NFS 系統會比較安全，不然會直接指定可以存取的 Client IP。

由於 NFSv3 和 NFSv4 Server 都是使用 exports 這個資源分享設定檔，而 NFSv4 多了虛擬根目錄的概念，所以若要兩個同時使用的話，要把所有分享的目錄 bind 到同一個子目錄下，故這時候 exports 的寫法要稍微注意一下，需符合 NFSv4 的規則，並且 bind 的規則也要記得寫到 fstab 裡面，這些會在接下來的 17.3 節中詳細介紹。

17.3 NFS Server 和 Client 程式安裝

安裝 NFS Server 只要輸入以下指令即可。

安裝 NFS Server :

```
sudo apt-get install nfs-kernel-server
```

安裝好 server 後，它會把 portmap，nfs-kernel-server 和 nfs-common 依序加到 runlevel 裡面，若要重新啟動 NFS 服務的話，也要依照這個順序來唷！

若只需要 client 端的功能，那如以下指令安裝。

安裝 NFS Client :

```
sudo apt-get install nfs-common
```

若要開機的時候預設不要自動啟用 NFS，請參考『深入瞭解 Ubuntu 系統』那一章，那裡對於 runlevel 的管理有仔細的介紹。

因為 NFSv3 和 NFSv4 的設定有一點不一樣，所以詳細設定的地方筆者就分開來講。讀者要使用 NFSv4 或者 NFSv3 就需要依照你的網路環境來評估。

表 17-3-1 NFS 相關的重要設定檔

功能	設定檔
設定 NFS Server 要分享的目錄與權限	/etc/exports
nfs-common 的啟動設定值 (NFSv4 需啟用 IDMAPD)	/etc/default/nfs-common
nfs-kernel-server 的啟動設定值	/etc/default/nfs-kernel-server
設定 client 端可以自動掛載 NFS Filesystem	/etc/fstab
idmapd 設定檔 (NFSv4 Only)	/etc/idmapd.conf

表 17-3-2 NFS 相關的重要指令

功能	指令
控制 NFS Server 啟動或關閉的 script 檔	/etc/init.d/nfs-kernel-server
控制 NFS Client 和 Server 共同的相關服務啟動或關閉的 script 檔	/etc/init.d/nfs-common
控制 portmapper 啟動或關閉的 script 檔。(NFSv4 可以不使用，但是 showmount 依然走 NFSv3 協定，所以建議打開吧)	/etc/init.d/portmap
修改 /etc/exports 後需重新更新分享目錄	exportfs -rv
察看本機目前分享的目錄	exportfs -v
查詢該主機有哪些目錄被分享出來	showmount -e <主機 IP>
查詢該主機有哪些 Client 端連上和主機分享資訊	showmount -r <主機 IP>

17.3.1 NFSv3 Server 的設定（主流穩定版）

設定 NFSv3 Server 最重要的就是 /etc/exports 這個設定檔，我們先弄個簡單的範例來看看吧！

Server 設定檔 /etc/exports 範例：

```
/home 192.168.0.0/24(rw,async,root_squash,no_subtree_check,insecure)
/tmp 192.168.0.0/24(rw,async,root_squash,no_subtree_check,insecure)
/opt 192.168.0.183(ro,root_squash,no_subtree_check,insecure)
```

如以上範例，exports 的格式是第一欄為要分享的目錄，第二欄是開放使用的 Client 主機，最後 (rw,async,.....) 裡面寫一些參數。在表 17-3-1-1 有 NFS Server 上使用的額外參數詳細說明，此範例中的參數是筆者最常用的。在這個例子中，Server 分享出 3 個目錄，前兩個讓 192.168.0.0 網段所有的電腦都可連進來，且都有讀寫的權限。最後一個只能由 192.168.0.183 那台機器連進來，並只能讀取。

當設定好後，讓它生效不需要重開服務，只要用以下指令即可！

重設 Server 分享目錄：

```
dbtsai@ubuntu:~ $ sudo exportfs -rv # 重設分享目錄清單
exporting 192.168.0.0/24:/home
exporting 192.168.0.0/24:/tmp
exporting 192.168.0.183:/opt

dbtsai@ubuntu:~ $ exportfs # 察看目前分享目錄，可加 -v 獲得更詳細資訊
/home 192.168.0.0/24
/tmp 192.168.0.0/24
/opt 192.168.0.183
```

Client 掛載 /home 範例：

```
# mount -t -o <掛載參數> nfs <Server IP>:<分享目錄> <mount point>
sudo mount -t nfs 192.168.0.180:/home /NFS/home

# 讀者也可以依照以下範例使用 soft mode 且改變讀取區塊大小為 8192。
sudo mount -t nfs -o soft,rsize=8192 192.168.0.180:/home /NFS/home
```

如 Client 掛載範例，我們只需要用以上指令就可以掛載上 Server 的遠端目錄到本機的 mount point 上，讀者可以自己用同樣的方式來掛載 tmp 和 opt。掛載時，也可以加上一些掛載參數，參數可以參考表 17-3-1-2，這些參數會在 fstab 那裡給出筆者最長用的設定，讀者可以參考看看。

若要看遠端有開放哪些資源可以提供掛載，可以使用以下方式。

察看遠端可分享的目錄資源：

```
dbtsai@ubuntu:~ $ showmount -e 192.168.0.183 # 可用 -a 來得知有多少 Client 連上
Export list for 192.168.0.183:
/export      192.168.0.0/24
/export/home 192.168.0.0/24
/export/tmp  192.168.0.0/24
```

表 17-3-1-1 NFS Server exports 重要的參數

參數	功能
rw	該分享目錄可以讀寫
ro	該分享目錄只能讀取
secure	不允許 Client 使用大於 1024 的 Client 端 port，也就是從 Server 傳遞資料到 Client 端的目的地 port 要小於 1024，此時 Client 端一定要使用 root 帳號才能 mount 遠端 NFS Server。通常會建議使用 insecure。
insecure	允許 Client 端自行決定自己機器使用的 port，筆者通常都會設這個，如此非使用 root 帳號的 client 端才能 mount NFS Server。
async	允許非同步寫入資料，所以效能會比同步寫入好。寫入時會先放到記憶體，等硬碟有空檔再寫入！風險為若 Server 當機或不正常關機，會損失資料，但通常 Server 都很穩定，所以建議打開獲得更好的效能！
sync	同步寫入，效能比較不好。
nohide	當 export 出兩個目錄，而其中一個目錄是另外一個目錄的子目錄，例如我們使用虛擬目錄的例子，此時我們 mount 根目錄時，會自動把所有子目錄 mount 起來。建議使用這個選項比較方便，尤其在 NFSv4 有虛擬目錄的情形。
hide	當 mount 根目錄時，export 出的子目錄需要自己明確的再掛載。
subtree_check	當分享的目錄是某個檔案系統的子目錄，選用這個可以確定父目錄的權限可以讓 NFS Server 分享使用。
no_subtree_check	剛好和上面的相反，因為不做權限測試，效能比較好。
fsid=0	定義 NFSv4 中的根目錄，只能有一個！

參數	功能
root_squash	在權限的地方有詳細解釋，代表當 client 用 root 會變成匿名權限
no_root_squash	關掉 root_squash 功能。
all_squash	在 Client 所有的使用者都是變成匿名身份
anonuid	指定在 Client 中操作該目錄的權限帳號，如使用 anonuid=0，那在 Client 端不管是誰都有著 root 的權限操作該目錄。
anongid	同上，不過這是指定使用者群組。

表 17-3-1-2 NFS Client 掛載重要的參數


參數	功能
rsize	每次讀取的大小，預設是 1024 byte，在區網內設大一點會有比較好的效能。但是若網路狀況不好時，設太大反而會導致丟包重傳而效能低下。在區網下，筆者建議使用 rsize=8192。(在 NFSv4 最大建議到 32768 byte)
wsize	每次寫入的大小，預設也是 1024 byte。在區網下筆者建議用 wsize=8192。(NFSv4 最大建議到 32768 byte)
proto=udp	使用 UDP 協定來傳輸資料，在區網中會有比較好的效能。若要跨越 Internet 的話，使用 proto=tcp 會有比較好的偵錯能力。
bg	掛載的時候使用背景模式，也就是第一次掛不上去時，丟到背景慢慢試。所以當 client 端掛載寫在 fstab 裡面，一定要用 bg，否則會拖慢開機速度。
fg	掛載的時候用前景模式，直到掛載上了才繼續跑別的程序。剛好和 bg 模式是相反的，這是預設值。
soft	當 NFS Client 以 soft 掛載 Server 後，若網路或 Server 出現問題，造成 Client 和 Server 無法傳輸資料時，Client 會一直嘗試到 timeout 後顯示錯誤並且停止嘗試。若使用 soft mount 的話，可能會在 timeout 出現時造成資料丟失，故一般不建議使用。
hard	這是預設值。若用 hard 掛載硬碟時，剛好和 soft 相反，此時 Client 會一直嘗試連線到 Server，若 Server 有回應就繼續剛才的操作，若沒有回應 NFS Client 會一直嘗試，此時無法 umount 或 kill，所以常常會配合 intr 使用。

參數	功能
intr	當使用 hard 掛載的資源 timeout 後，若有指定 intr 可以在 timeout 後把它中斷掉，這避免出問題時系統整個被 NFS 鎖死，建議使用。


17.3.2 NFSv4 Server 的設定（實驗性質）

雖然 NFSv4 有很多新的功能，但是在 kernel 中還是被標記為實驗用，所以你可能遇到一些莫名其妙的 bug，例如筆者更新核心到 2.6.17 後，發現 NFSv4 Client 會莫名的當掉，最後再更新到 2.6.18rc4 就正常了！而 kernel 裡面關於 NFSv4 的程式碼又會比原始研發單位還要慢一點，若你想要用最新的 NFSv4 程式碼，可以到 <http://www.citi.umich.edu/projects/nfsv4/linux/> 抓最新的 patch。若讀者希望一切都沒問題，不想再做一些 hack，可以考慮用穩定的 NFSv3。但若要在 Internet 傳輸，那麼 NFSv4 的容易設定防火牆與加密的特性可以考慮一下！

由於 NFSv4 使用 rpc.idmapd 這個 daemon 來處理 User ID、Group ID 和權限的對應，所以這個 daemon 在 Client 端和 Server 端都要啟動，並且需要自己手動建立通訊 pipe 給 svc gssd subsystem 用，這些在 Ubuntu 都需要手動設定。記得這些設定一定要在 Server 和 Client 分別做過一次唷，設定好了就不需要再做了！

 建立 pipe 目錄給 NFSv4 使用：


```
sudo mkdir /var/lib/nfs/rpc_pipefs
```

 加入以下兩行到 /etc/fstab 讓 rpc_pipefs 和 nfsd fs 在開機自動 mount 起來：

```
rpc_pipefs /var/lib/nfs/rpc_pipefs rpc_pipefs defaults 0 0
nfsd /proc/fs/nfsd nfsd defaults 0 0
```


 由於我們不重新開機，為了讓剛剛加入的兩行生效，執行以下指令吧！：

```
sudo mount rpc_pipefs
sudo mount nfsd
```


 編輯 `/etc/default/nfs-common`，把 `NEED_IDMAPD` 改成 `yes`：

```
NEED_IDMAPD=yes
```

接下來重新啟動 `portmap`，`nfs-kernel-server` 和 `nfs-common`，要依照筆者下面列出來的順序操作喔！

 對於 Server 需要重新啟動這三個服務：

```
sudo /etc/init.d/portmap restart
sudo /etc/init.d/nfs-kernel-server restart
sudo /etc/init.d/nfs-common restart
```

 對於 Client 需要重新啟動這兩個服務：


```
sudo /etc/init.d/portmap restart
sudo /etc/init.d/nfs-common restart
```

這時候可以開始設定 `/etc/exports`。基本上和 NFSv3 差不多，只不過多了虛擬根目錄的概念。NFSv4 為了安全性問題，引入了 Pseudo Filesystem 的機制，這類似 HTTP Server 是從系統上的某個子目錄當成網頁的起始目錄(虛擬根目錄)。

 設定檔 `/etc/exports` 範例一 (不正確的設定)：

```
/home 192.168.0.0/24(rw,fsid=0,async,no_root_squash)
/tmp 192.168.0.0/24(rw,async,no_root_squash)
```

如以上範例，筆者把 `home` 和 `tmp` 分享出來，先注意到 `fsid=0` 的參數，這個參數在 NFSv4 是有意義的！這宣告該分享目錄是根目錄，所以在 NFSv4 Client 端要 `mount` 時，需要 `mount -t nfs4 IP:/ /mnt/nfs`，而不是傳統的 `mount -t nfs IP:/home /mnt/nfs`。所以在 NFSv4 和 NFSv3 最大的不同是，在 v4 要先用 `fsid=0` 定義出一個根目錄，也只有那個根目錄以下的子目錄才能被掛載，筆者會在接下來用範例來解釋。而虛擬根目錄只能定義一個，若定義多個的話，會造成不可預期的問題。

 Server 重設分享目錄：

```
dbtsai@ubuntu:~ $ sudo exportfs -rv
```

Client 掛載 /home 範例：

```
sudo mount -t nfs4 192.168.0.183:/ /NFS      # 用 NFSv4 協定掛載
sudo mount -t nfs 192.168.0.183:/home /NFS  # 用 NFSv3 協定掛載
```

讀者應該有注意到吧！掛載同一個 Server 上的 /home 目錄，因為在 Server 上的 exports 有指定 /home 是虛擬根目錄（fsid=0），所以 NFSv4 遠端掛載點是和 NFSv3 不一樣的喔！在我們這個範例中被 exports 出來的 tmp 將無法在 NFSv4 中被掛載，因為它不在虛擬根目錄底下阿！所以使用 NFSv4 時，我們常常會建立一個根目錄，然後把其它要分享的目錄資源用 mount --bind 掛載進來，如以下範例。當然大家不會希望每次都要做 bind 的動作，所以請參考 17-3-3 小節把設定寫到 fstab 吧！

使用 bind 來建立虛擬根目錄範例：


```
sudo mkdir /export      # 建立 NFSv4 分享的虛擬根目錄
sudo mkdir /export/home # 由於所有 NFSv4 分享目錄都要在虛擬根目錄底下，
sudo mkdir /export/tmp  # 所以建立 /home 和 /tmp 來給 bind 用！
sudo mount --bind /home /export/home
sudo mount --bind /tmp /export/tmp
```

設定檔 /etc/exports 範例二 (正確的設定)：

```
/export
192.168.0.0/24(rw,fsid=0,nohide,insecure,no_subtree_check,async,root_squash)
/export/home
192.168.0.0/24(rw,nohide,insecure,no_subtree_check,async,root_squash)
/export/tmp
192.168.0.0/24(rw,nohide,insecure,no_subtree_check,async,root_squash)
```

改成這樣後，我們先 sudo exportfs -rv 讓新的 exports 生效，這已經解決虛擬根目錄的問題，因為我們使用 bind 來連接 /home 到 /export/home，也就是這兩個目錄是等義的，這樣規劃符合 NFSv4 的單一虛擬根目錄的原則，同時也可以在 NFSv3 中被掛載！

不論 Client 是 NFSv3 或 NFSv4，在參數加上 nohide，這樣 Client 端掛載一個目錄時，會自動掛載該目錄下其他可掛載資源，而不需要明確的指定並一個一個掛上。

 Client 掛載 /export 範例：

```
sudo mount -t nfs4 192.168.0.183:/ /NFS # 用 NFSv4 協定掛載
sudo mount -t nfs 192.168.0.183:/export /NFS # 用 NFSv3 協定掛載
```


如以上範例，當 Server 端有設 nohide 時，Client 在 NFSv4 會自動掛載 /tmp 和 /home 或在 NFSv3 會自動掛載 /export/home 和 /export/tmp，不需要再另外設定。

當然也可以直接掛載子目錄，如以下範例。

 Client 掛載 /export/home 範例：

```
sudo mount -t nfs4 192.168.0.183:/home /NFS # 用 NFSv4 協定掛載
sudo mount -t nfs 192.168.0.183:/export/home /NFS # 用 NFSv3 協定掛載
```

若讀者使用 showmount 去看 Server 分享的資源時，要注意的是這個方式顯示的目錄是走 NFSv3 舊的協定，所以沒有考慮到虛擬根目錄。故使用 NFSv4 掛載法需要自己把 /export 去掉喔！

 察看遠端可分享的目錄資源：

```
dbtsai@ubuntu:~ $ showmount -e 192.168.0.183 # 可用 -a 來得知有多少 Client 連上
Export list for 192.168.0.183:
/export      192.168.0.0/24
/export/home 192.168.0.0/24
/export/tmp  192.168.0.0/24
```

或許讀者會覺得奇怪，不是說 NFSv4 不需要 portmap 了嘛？那我們怎還把它啟動呢？的確是不需要，若你把它關了是可以正常連線，但 showmount 是走舊的協定，所以就沒辦法看 Server 分享的資源啦！所以若你用不到 showmount 且不用 NFSv3，可以把 portmap 關了。

17.3.3 將設定寫到 fstab 來自動掛載吧！

在 Client 中把掛載設定寫到 /etc/fstab 裡面，讓每次開機都可以自動掛載遠端的 NFS Server 是很方便的。在 Server 中若使用到 NFSv4，常常用 bind 的技巧

製造一個虛擬根目錄，這些 bind 的規則也經常寫到 /etc/fstab 裡面。這裡的討論繼續沿用前兩小節的範例，所以讀者要視需求修改唷！

筆者先討論如何在 Server 上開機後自動把虛擬目錄和真正的分享目錄 bind 起來。首先編輯 Server 上的 fstab 吧！

Server 上的 /etc/fstab 範例：

```
# <來源目錄>    <目的地目錄>        none    bind    0    0
/home           /export/home          none    bind    0    0
/tmp           /export/tmp            none    bind    0    0
```

如以上範例，在 Server 上的 fstab 最後面加上這兩行，開機後就會自動把你真正要分享的目錄 bind 到虛擬路徑底下唷！

在 Client 端要設定自動掛載也是加到 fstab 裡面，如以下範例。

Client 設定 /etc/fstab 來自動掛載 NFS Server：

```
# <Server IP>:<分享目錄>    <掛載點>    nfs4 或 nfs    <參數>    0    0

# 使用 NFSv4 協定掛載
192.168.0.183:/              /NFSA  nfs4  proto=udp,rsize=32768,wsiz=32768,intr 0  0

# 使用 NFSv3 協定掛載
192.168.0.183:/export       /NFSA  nfs   proto=udp,rsize=8192,wsiz=8192,intr 0  0
```

這個範例中，Client 在開機後會自動 mount Server 上的 /export。不過一個是用傳統的 NFSv3，另外一個是 NFSv4，它們可以同時使用。再次提醒，它們掛載的分享目錄不同是因為 NFSv4 有虛擬根目錄的概念。後面的參數讀者可以參考使用，筆者測試這樣的參數在區網效能會比較好，參數詳細說明在表 17-3-1-2。

若你修改過 fstab 後，不想要重新開機，可以透過以下指令重新依照 fstab 的內容掛載。

依照 fstab 的內容重新掛載：

```
sudo mount -a
```

若你要看目前系統上有哪些 File system 被掛載上來，可以透過 mount，它也會列出本地硬碟的掛載情形。

瞭解目前有哪些檔案系統被掛載：

```
dbtsai@ubuntu:~ $ mount
...(略) ...
nfsd on /proc/fs/nfsd type nfsd (rw)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
192.168.0.183:/ on /NFS type nfs4 (rw,proto=udp,intr,addr=192.168.0.183)
```

若當你要卸載 NFS Filesystem，遇到系統資源忙碌而無法 umount，這時候用 Lazy umount 很有用。它會偵測該裝置是否忙碌，等到有空檔就自動幫你 umount，通常筆者會配合 force umount 使用。

卸載 NFS Filesystem：

```
sudo umount -l -f /NFS # umount /NFS 這個 NFS Filesystem
```

看到這裡，讀者的 NFS 系統應該也架起來了吧！若有問題，可以先檢查一下防火牆設定是不是把它擋掉了，若真的找不到原因，也歡迎和筆者聯絡，在能力所及的範圍下可以互相討論。

CHAPTER 18

FTP 伺服器

這個章節，因為筆者來不及寫，所以委託筆者的好友 Haway 來幫忙，也感謝 Haway 抽空寫了這個章節。不然就真的會趕不上交稿的日期了。

18.1 基礎 FTP 概念

在網路上我們要與別人分享檔案，除了網頁伺服器之外，還有一種就是檔案傳輸協定（File Transfer Protocol），簡稱 FTP。利用 FTP 來傳輸檔案，可以很容易在不同平台之間分享檔案，不論是 Linux、Windows 或是 Mac。剛開始先來認識一下什麼是 FTP 協定：當我們在網路上瀏覽時，在網址部份是 `http://` 開頭的是屬於網頁伺服器所提供的服務，而 `ftp://` 開頭的位置，就是檔案伺服器所提供的服務，瀏覽器會依照不同的網址開頭而去連接不同的伺服器，所以簡單的說我們只要看網址的開頭，就可以知道目前是使用那一種協定。如果你是使用 `ftp client` 的軟體，例如 `filezilla`、`cuteftp` 等，那就一定是使用 `ftp` 協定，所以不需要 `ftp://` 開頭。

了解 FTP 之前，先來介紹一下 TCP/IP 的大略運作，TCP/IP 可是一門深奧的學問，花兩三個篇幅說完那是不可可能的，它的學問可以出一整本的書了！所以這裡筆者只敢說：我們「大略」來說一下 TCP/IP 吧！^_^。每一台電腦，連上網際網路，都會配置到一個 IP 這個 IP 是獨一無二的，而每一個 IP 都會有 65535 個埠（port），從不同的埠進入就會對應到不同的服務，打個比方，如果網際網路就像一條大馬路，大馬路旁邊會有很多住家，每一個住家都會有獨一無二的住址，你可以在你家門面裝 65535 個門，每一個門進入後會走到 65535 個相對應的房間。在這裡，大馬路就像網際網路，住家就像我們的個人電腦，住址就是 IP，每一個門就像一個埠，每一個房間就像一個服務，所以你從一個 IP 位置進入，走入相對應的埠，就可以連接到一個服務，例如連線位置 IP：192.168.1.2 埠：80 表示連線到 192.168.1.2 這台機器的 http 伺服器，而 FTP 伺服器就是埠 20 和 21，小於 1024 的埠，都有規定用途，不可亂用，通常架設伺服器會使用預設的埠，而自行設定的話都會使用大於 1024 以上的埠。

當我們開啟一個 FTP 連線的時候，如圖 18.1，我們的電腦會自動幫我們建立兩條連線，注意喔！是兩條連線，這跟一般的網頁瀏覽是不一樣的，網頁瀏覽器只會利用 Port 80 來建立連線，而 FTP 連線會由客戶端開啟大於 1024 的埠來連接伺服器 20、21 兩個埠來傳輸，21 埠稱為命令埠，20 埠稱為資料埠，先連接的是命令埠，傳送的是 ftp 的指令，例如我們在操作 ftp 的時候有「上一頁」、「瀏覽目錄清單」、「改變工作目錄」等，這類型的動作都是在命令埠傳送；而清單目錄，還有我們要傳資檔案的時候，就是透過資料埠傳送是由伺服器主動連線到客戶端，所以兩個埠都必須開啟，才能啟用一個正常的 ftp 連線，很多人在設立防火牆的時候會忽略了 20 埠而沒有開啟，在此希望大家多注意。

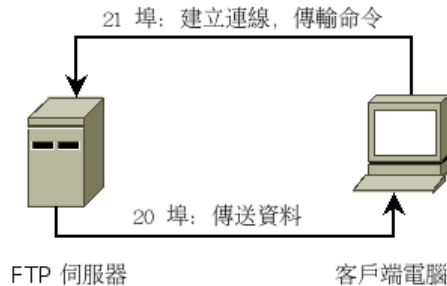


圖 18.1 FTP 連線

18.2 主動 (Active) 連線和被動 (Passive) 連線

上面介紹了 ftp 的一般連線是透過兩個埠，圖 18.1 所示是主動連線，客戶端電腦會用亂數開啟大於 1024 埠然後連接到伺服器的 21 埠，當客戶端準備接收資料的時候，會再開啟一個大於 1024 的埠，當然也是亂數決定的，並且告知伺服器，伺服器就會主動透過自己的 21 埠連接到客戶端的埠，這是**主動連線**。但現在很多客戶端都會放在 NAT 伺服器底下，透過轉址的方式對外連線，這樣就會造成問題，主要是因為當我們透過 NAT 轉址的時候，實際連到伺服器的位置是 NAT 主機的位置，而命令埠是由客戶端連線到伺服器沒有問題，資料埠是由伺服器端連接到客戶端就會產生連接不到的問題，因位伺服器端會嘗試連接 NAT 主機，而不是客戶端主機，如圖所示。

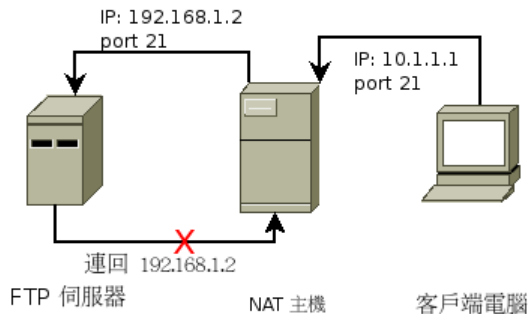


圖 18.2 NAT 下的連接錯誤

解決的方法有兩種，分別在 NAT 主機或是客戶端則一即可，若是你有權力操作 NAT 主機，而剛好又是 Linux 系統，那只要將 `ip_conntrack_ftp` 與 `ip_nat_ftp` 這兩個模組載入即可，若是你剛好不熟 Linux 或是沒有操作 NAT 主機的權力，那可以嘗試在 FTP 客戶端軟體中開啟被動連線即可。剛剛有提到在主動連線的時候第一個開啟的連線是由客戶端主動連接到伺服器的命令埠，而這樣通過 NAT 的連線就不會有問題，我們如法炮製將資料埠也改成由客戶端連線到伺服器，這是**被動連線**。第一次的連線依舊，而當我們要傳遞資料的時候，客戶端會告知伺服器使用 Passive 連線，伺服器會開啟一個大於 1024 的埠，並且等待客戶端連線，這時客戶端就可以透過 NAT 主機正常的連線到伺服器。



圖 18.3 Active 連線模式

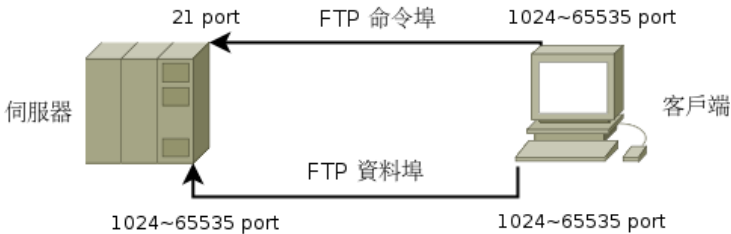


圖 18.4 Passive 連線模式

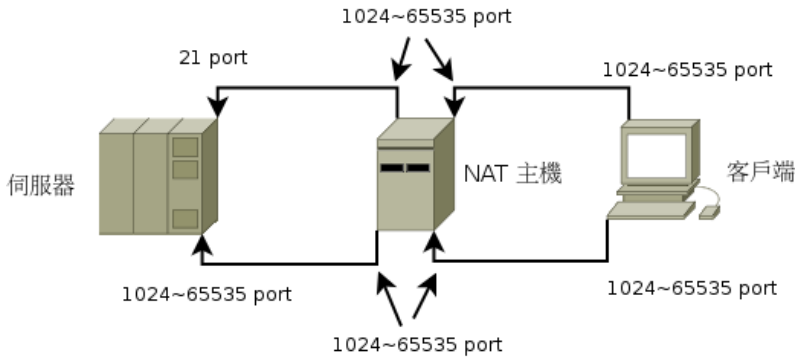


圖 18.5 Passive 穿越 NAT 主機的連線

18.3 FTP 伺服器

那我們如何架設一個 FTP 伺服器呢！首先，先瞭解一下目前在 Linux 上面有哪些伺服器軟體可以安裝，在依照不同的需求與特性來選擇你自己想要的軟體。目前 Linux 上面比較著名的大約有：

1. Proftpd 官方網頁 <http://www.proftpd.org/>
2. vsftpd 官方網頁 <http://vsftpd.beasts.org/>
3. wuftpd 官方網頁 <http://www.wu-ftp.org/>

當然還有其他很多的軟體，每個的功能與操作都不盡相同，在此不列舉以免浪費頁數，如果有興趣可以在網路上找找資料應該會有很多，到哪裡找？應該不用說了吧！^_^。這裡我們拿 proftpd 來為各位作示範如何架設一個檔案伺服器，先安裝起來吧！

18.3.1 安裝、登入 FTP 伺服器

在 ubuntu 底下有提供 Synaptic 這個軟體來管理所有安裝在我們電腦裡的軟體，執行系統→管理→Synaptic 套件管理程式，或者是打開終端機，執行 `sudo synaptic`，接著系統會要求你輸入密碼切換成 root 身份來安裝系統。

```
$ sudo synaptic
Password:
```

就會跳出如下圖的畫面



圖 18.6 Synaptic

可以利用上方的「搜尋」按鈕找尋比較快，輸入 proftpd 就可以找到主要的程式與其他相關套件。



圖 18.7 搜尋套件

搜尋後的結果大約有：

1. proftpd 主程式
2. proftpd-common 管理程式
3. proftpd-doc 說明文件
4. proftpd-ldap 支援帳號認證使用 LDAP
5. proftpd-mysql 支援帳號認證使用 MySQL 資料庫
6. proftpd-pgsql 支援帳號認證使用 PostgreSQL 資料庫

筆者建議最少也要安裝 proftpd、proftpd-doc 這兩個套件，如果有需要特殊的帳號認證方式，就再安裝其他的帳號支援套件。

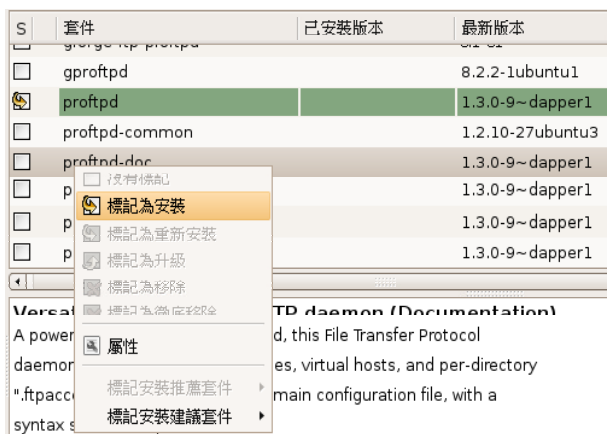


圖 18.8 安裝 Proftpd

標記完成之後，點選上方的「套用」，會跳出確認視窗，按下三角形可以確認目前有哪些軟體需要安裝，再次點選「套用」，就可以將 Proftpd 給裝好了。

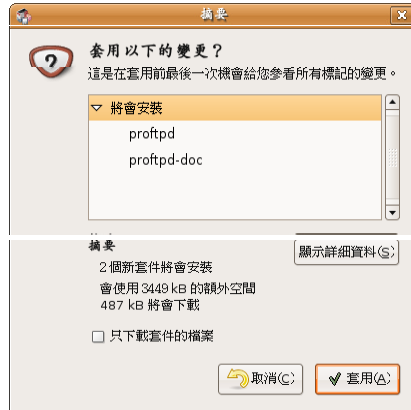


圖 18.9 安裝 Proftpd

接著，會跳出 Proftpd 的設定視窗，點選下一頁整個系統就會開始安裝。

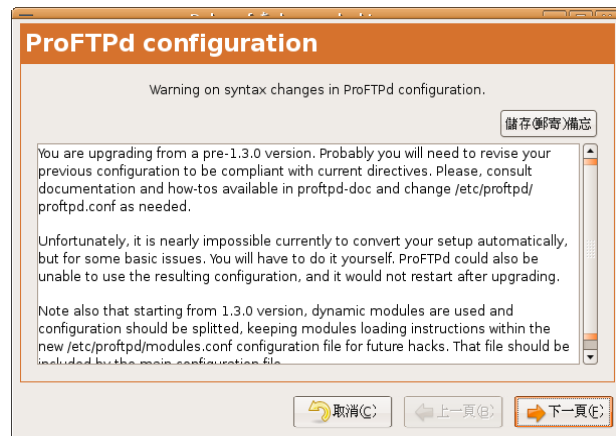


圖 18.10 Proftpd 設定視窗

夠簡單吧！上述是圖形介面的安裝方式，如果你的 ubuntu 只有安裝文字介面，那怎麼辦呢？首先你登入系統後，請切換到 root 身份：

```
$ sudo su -
Password :
```

之後可以利用 apt 的相關指令進行安裝。如用 apt-cache 可以尋找軟體，apt-get 可以安裝軟體，當然我們直接拿要安裝的 Proftpd 示範。

```
# apt-cache search proftpd
proftpd - Versatile, virtual-hosting FTP daemon
proftpd-doc - Versatile, virtual-hosting FTP daemon (Documentation)
proftpd-ldap - Versatile, virtual-hosting FTP daemon (dummy transitional package)
proftpd-mysql - Versatile, virtual-hosting FTP daemon (dummy transitional package)
proftpd-pgsql - Versatile, virtual-hosting FTP daemon (dummy transitional package)
ftpd - FTP server
gforge-ftp-proftpd - Collaborative development tool - FTP management (using ProFTPd)
gproftpd - GTK+ configuration tool for proftpd
proftpd-common - Versatile, virtual-hosting FTP daemon
```

找到我們要安裝的軟體之後，就可以用 apt-get install 來安裝。

```
# apt-get install proftpd
讀取套件清單中... 完成
了解套件依存關係中... 完成
[ ... 略 ... ]
正在預先設定套件 ...
選中了曾被取消選擇的套件 proftpd。
(正在讀取資料庫 ... 系統目前總共安裝有 81087 個檔案和目錄。)
正在解壓縮 proftpd (從 .../proftpd_1.3.0-9~dapper1_i386.deb) ...
正在設定 proftpd (1.3.0-9~dapper1) ...
正在新增系統使用者「proftpd」...
正在新增群組為「proftpd」的新使用者「107」(nogroup)。
未建立家目錄「/var/run/proftpd」。
正在新增系統使用者「ftp」...
正在新增群組為「ftp」的新使用者「109」(nogroup)。
正在建立家目錄「/home/ftp」。
「/usr/share/proftpd/templates/welcome.msg」 ->
「/home/ftp/welcome.msg.proftpd-new」
* Starting ftp server proftpd - IPv6 getaddrinfo 'localhost' error: Name or service not known
[ ok ]
```

筆者將安裝中的一些訊息忽略了，只留下一些重要的訊息讓讀者閱讀，中間可以看到系統會幫你自動增加一個使用者與群組，名稱都是 proftpd，並且幫你建

立一個 /home/ftp 的目錄，每一個 Linux 套件都不全然相同，有些並不會幫你建立使用者或是目錄，要隨機應變，^_^，最後一行有一個訊息是：

```
Starting ftp server proftpd - IPv6 getaddrinfo 'localhost' error: Name or service not known
```

這行表示這台主機沒有使用 IPv6，所以回應這個訊息給你，沒關係！我們有 IPv4 就夠了。安裝好之後，你可以使用先前介紹的 Filezilla 來登入測試一下，在 Host 的地方輸入 localhost 以及自己的帳號密碼，就可以登入剛架設好的伺服器了！開心吧～

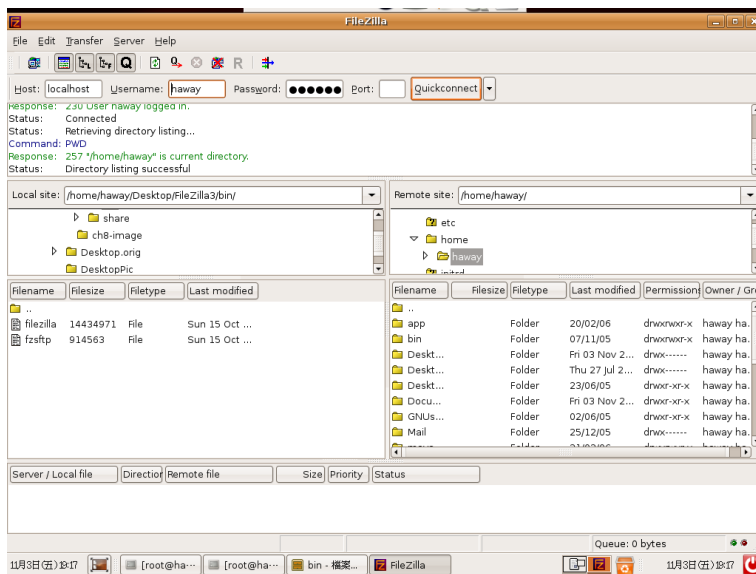


圖 18.11 測試 FTP 伺服器

最基礎的 ProFTPD 我們已經架設好了，你應該也用自己的帳號跟密碼登入成功，如果你想開放給其他人使用，只需要在系統幫他建立一個帳號，並且給他你的 IP 位置，他就可以連線進來了，連線進來後會登入家目錄中。對於要架站的人來說，請盡量使用固定 IP 來架站，因為浮動 IP 會在你每一次重新開機的時候變動 IP 位置，那你就又要重新一個一個告訴你的好朋友了。

如果你的 FTP 想讓任何人都可以進來，這種稱為匿名登入，例如你想要抓取 Linux 套件的時候，可以登入 <ftp://ftp.isu.edu.tw> 義守大學的 FTP 伺服器，但不

需要帳號與密碼。要開啟 ProFTPD 的匿名登入，很簡單，你只要編輯 ProFTPD 的設定檔就可以了，路徑位於 /etc/proftpd 底下，執行：

```
$ sudo gedit /etc/proftpd/proftpd.conf
```

如果你習慣 vi，把上面指令的 gedit 換成 vi 就好了，打開 proftpd.conf 之後，把游標移到最下面，有一個區塊是類似 <Anonymous ~ftp> 開頭並以 </Anonymous> 結尾的，這個區塊的設定就是專門為匿名登入的連線設置，只要是匿名登入，全部都會受限於這裡的設定，我們來看幾的比較重要的設定（以 # 開頭的表示註解）：

```
<Anonymous ~ftp>
  User ftp
# 匿名登入的身份：所有匿名登入的連線，都會用這個 ftp 使用者的身份。
# 你可以用 cat /etc/passwd | grep ftp 這個指令來檢查 ftp 使用者的相關設定。
  Group nogroup
# 匿名登入的群組：所有匿名登入的連線，都會用這個 nogroup 這個群組。
  UserAlias anonymous ftp
# 使用者別名：當登入身份是 anonymous 的時候等同於 ftp 這個身份。
  RequireValidShell off
# 檢查登入者身份是否有正確的 Shell：請用 cat /etc/passwd | grep ftp 看最後一個 # Shell
# 的設定是否存在 /etc/shells，off 表示不檢查。
  MaxClients 10
# 匿名登入的最大連線數量：限制同時能匿名登入的人數。
  # Limit WRITE everywhere in the anonymous chroot
  <Directory *>
    <Limit WRITE>
      DenyAll
    </Limit>
  </Directory>
# 這裡是很重要的設定，表示所有匿名登入（也就是 ftp 身份）的人，不能寫入任何資料到主機，
# 也就是不能上傳資料，這裡是安全性的設定，若非特殊情況，請不要給匿名的帳號有寫入的權限，
# 因為這樣駭客可以透過寫入執行檔來入侵電腦，切記！！
  # <Directory incoming>
  # # Umask 022 is a good standard umask to prevent new files and dirs
  # # (second parm) from being group and world writable.
  # Umask 022 022
  # <Limit READ WRITE>
  # DenyAll
```

```

#         </Limit>
#         <Limit STOR>
#         AllowAll
#         </Limit>
# </Directory>
# 如果一定要開啓匿名身份可以寫入，那就將上面這區塊的設定打開（去除最前面一個 #），
# 這段是設定 incoming 資料夾是可以寫入的，這個 incoming 資料夾位於 /home/ftp
# 目錄底下，如果你想自己建立目錄，例如 update，那就改成 <Directory update>
# 就可以了。建立/home/ftp/update 目錄之後，別忘了將目錄擁有者與權限設定正確喔。
</Anonymous>
# 匿名區塊結束。

```

18.4 ProFTPD 主要設定

奇怪，一般人都會先講解主要設定，後面才會說明匿名登入的，筆者個性怪怪的，請各位不要介意 ^_^。剛剛用 gedit 打開了設定檔，相信你對上半部的設定也是非常有興趣的吧！一樣來講解幾個重要的設定吧！

```
ServerName          "uBuntu"
```

ServerName 設定的是當 Client 登入的時候，要顯示的站台資訊，當你用 Filezilla 登入 ProFTPD 的時候，不知道有沒有注意看那一串跑很快的訊息？第一行就是跟這個設定有關，這裡我們設定成 uBuntu，請看圖

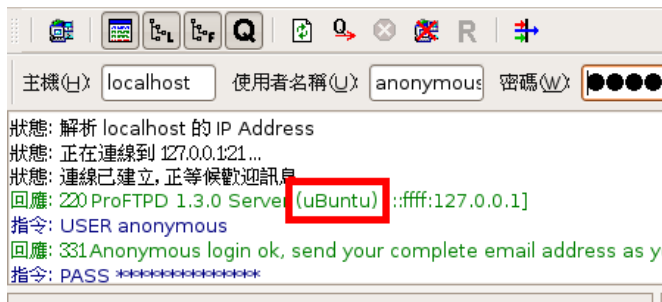


圖 18.12 ServerName 設定

```
ServerType          standalone
```

在 Linux 底下，所有的 daemon 程式有兩種啟動的方法，第一種是服務本身一直處於執行狀態，當有連線進來的時候就會馬上接手，稱為 Standalone 模式，

另外一種是服務本身是「睡著」的，而會有一個代理程式 xinetd 代為接收連線，xinetd 收到連線的時候會「叫醒」ProFTPD 然後轉手將連線交給 ProFTPD，稱為 inetd 模式，這樣有什麼不一樣呢。Standalone 的模式程式會一直處於執行狀態，會佔用記憶體跟系統資源，但是有連線進來的時候可以很快速的處理。inetd 模式會透過 xinetd 這個程式來喚醒其他程式，所謂的其他程式的意思就是 xinetd 可以同時管控很多個 daemon，如 proftpd、sshd 等，而只需要佔用一份資源（就是 xinetd 本身），只有在需要的時候才會喚醒相對應的程式，但也因為這樣，所以連線的時候會稍慢。如果你不知道你的站台要用哪種模式執行，那筆者給你的建議是，流量大的站台使用 Standalone 模式，因為要隨時處理新連線，而流量小的站台，就使用 inetd 模式，可以減少系統資源。

```
DisplayLogin      welcome.msg
```

如果你想在別人登入站台的時候顯示一些資訊，那就將內容寫入家目錄底下的 welcome.msg 裡面，在登入的時候就會顯示在那一串跑很快的訊息中，筆者建議放一些歡迎詞就好，因為這些訊息顯示得太快了，通常大家都不會注意的。

```
Port              21
```

這是一個很重要的設定，前面我們有提到關於「埠」的問題，這個設定就是說明了 ProFTPD 要接收那一個埠的設定。一般來說開的埠都是 21 埠，但也因為這樣，很多駭客都會刻意去連線 21 埠來嘗試入侵電腦，所以有些人會將他修改成 9999 或是自己喜愛的號碼，但修改之後要記得告訴別人你開放的埠喔！

```
PassivePorts     49152 65534
```

在被動模式的時候，是 Server 會開啟一個隨機的埠，等待 Client 連線，此行設定這個隨機所開啟的範圍。有些 Server 會搭配防火牆來設定此參數。

```
User             proftpd
Group            nogroup
```

執行 ProFTPD 的時候的身份與群組。

```
Umask            022 022
```

設定 Umask 為 022。

```
AllowOverwrite   on
```

此參數設定為 on 表示當傳輸檔案已存在的時候，會直接覆寫不提醒。

```
RootLogin          off
```

不允許 root 這個帳號登入伺服器，這個參數若是沒有的話，自行加上即可，可以避免駭客利用 FTP 來猜 root 的密碼。

```
DefaultRoot       ~ Ducati
```

這個參數表示 Ducati 這個**群組**，是群組喔不是使用者，會進入「chroot」環境，只能在自己的家目錄底下活動，無法離開家目錄，可以將使用者「關」在一個範圍裡面。可以用 ! 來反向操作。

其他當然還有很多設定，從 <http://www.proftpd.org> 裡面可以查詢到相關的功能，各位讀者要多加利用喔！

修改完設定檔之後，別忘了重新啟動伺服器。

```
$ sudo /etc/init.d/proftpd restart
```

確定 ProFTPD 已經在執行。

```
$ netstat -l | grep ftp
tcp6      0      0  *:ftp          *:*            LISTEN
unix  2      [ ACC ]     STREAM        LISTENING      38885    /var/run/proftpd/proftpd.sock
```

18.5 UTF8 和 Big5

UTF8 跟 Big5 檔案編碼的大戰，目前在 Linux 上面可以說是如火如荼啊！現在可以說是 utf8 跟 Big5 的轉換期。Big5 是早期的中文編碼，而 utf8 是新的中文編碼，早期的 Linux 都一律採用 Big5 的編碼，而 utf8 是未來的趨勢，像筆者所安裝的 Ubuntu 6.06 版的預設編碼是 utf8，而筆者有之前架設的 FTP 伺服器，上面就是採用 Big5 編碼，這樣聽起來或許不是什麼問題，那我們來試試看吧！筆者用其他 FTP Client (gftp) 從 utf8 的 ubuntu 登入 big5 編碼的 FTP 伺服器，會出現如下的畫面。

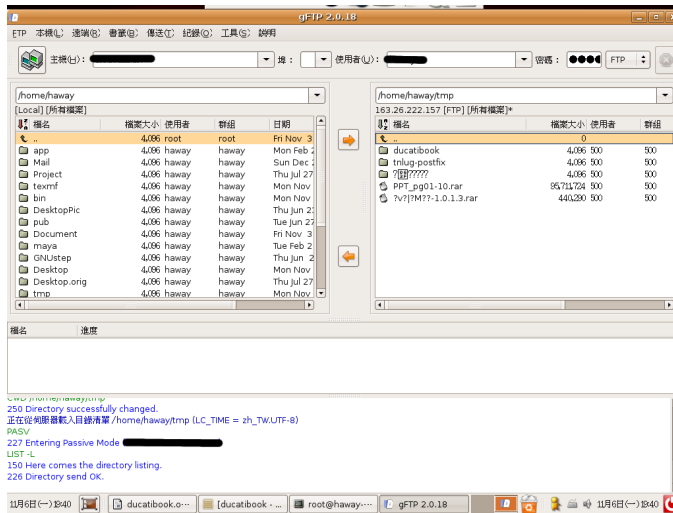


圖 18.13 不同編碼的 FTP 伺服器

你可以清楚的看到右方的 Server 端清單中文出現了亂碼的情況。這就是雙方編碼不一致所產生的問題，而 Filezilla 可以輕鬆解決不同編碼所帶來的問題，像上述的問題，你必須先確定 Server 伺服器的編碼，有一個很簡單的測試方法，就是利用 Firefox 來登入 FTP 伺服器，然後點選檢視→語系及字元編碼。看這裡的設定是否能正確顯示清單，我們利用 Firefox 再次登入上面的伺服器，然後檢視編碼。

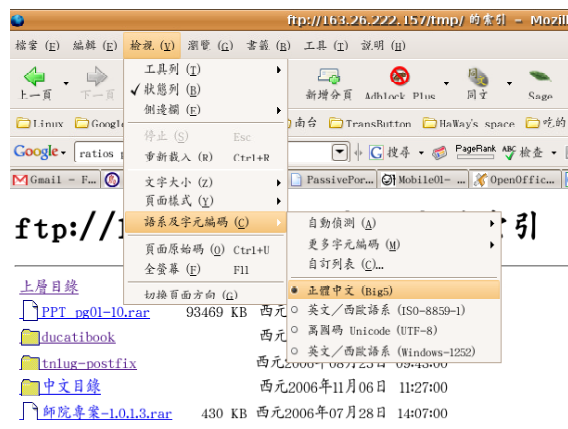


圖 18.14 利用 Firefox 確認編碼

就可以確認 Server 端的編碼是 Big5 的，開啟 Filezilla 之後點選檔案→站台管理員，在此新增我們最常用的 FTP 站台。



圖 18.15 Filezilla 站台管理員

最後有一個籤頁是「字碼集」，在此我們可以將 Server 端的編碼指定為 Big5，之後再次登入就可以正常顯示啦，如果沒設定的話，是連中文都不會顯示的喔！！



圖 18.16 Filezilla 字碼集

再次登入後就可以正常顯示中文了！

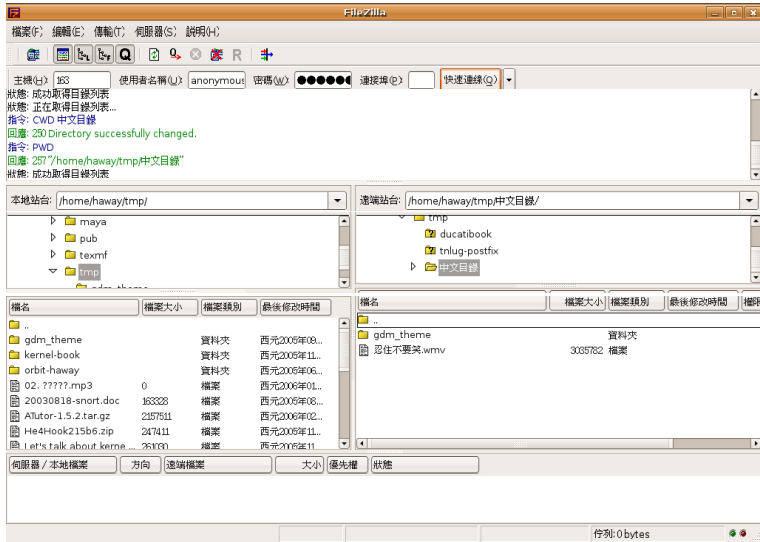


圖 18.17 Filezilla 中文沒問題啦！

CHAPTER 19

圖型化界面的防火牆與 iptables 概觀

說到了防火牆，讓我想到了在 Ubuntu 上，筆者曾用過一個 GUI Firewall：Firestarter，說真的蠻方便的，而且蠻適合一般人用的，透過圖型介面的方式，就可以修改底層的 iptables 規則。不過在使用前，讀者還是必需對底層的 iptables 有些了解。而這個章節其實原本是來不及寫的，所以筆者就委託了好友「企鵝狂」幫忙寫，企鵝狂也很熱心一口就答應了，在這裡筆者要感謝一下企鵝狂的熱心，讓這本書更完整了。那麼就我們來看看 Firestarter 倒底有什麼好料的呢？

19.1 Linux 防火牆功能

Linux 網路功能中最強大的特色就是內建了 iptables/netfilter 防火牆的機制。netfilter 是 Linux 核心中防火牆的架構，iptables 則是用來管理防火牆規則的工具。那 Linux 的防火牆可以做到甚麼功能呢？任何你能想的出來的，還有其他你想不出來的。包含封包過濾、封包修改、紀錄特定封包等等。以下先簡單介紹一下 Linux 核心裡面封包處理的流程，之後在介紹如何使用 GUI 工具 Firestarter 來建立簡單的防火牆。

Kernel 中的封包處理流程

網際網路中，一個封包從主機 A 發出之後，可能經過許多中間的路由器之後才會到達終點 B。

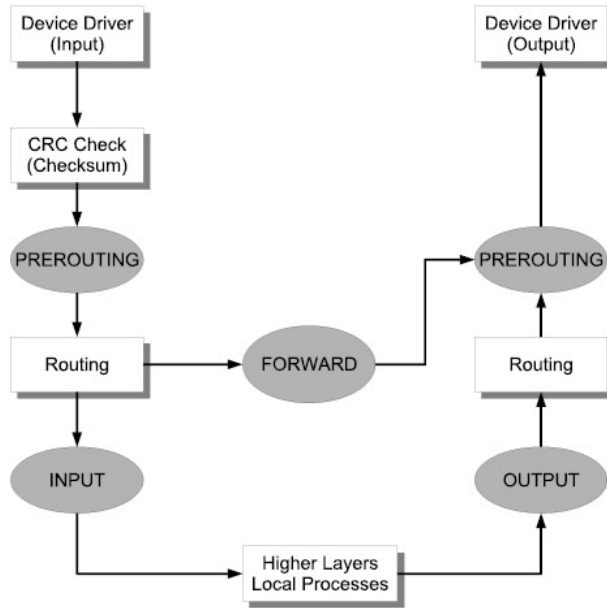


圖 19.1 封包的處理流程

當一台主機收到了網路的封包，會檢查這個封包的目的地是不是本機位址，如果是，封包就會經過 iptables INPUT Chain 來檢查，然後交給 OSI 網路分層中更上層的通訊協定來處理。如果不是，那就會經由 FORWARD 以及 POSTROUTING Chain 來檢查，然後轉送出去。

從本機發送出去的封包，則會經過 OUTPUT 及 POSTROUTING Chain 的檢查然後發送。

為了讓 iptables 規則的管理更清楚簡便，iptables 分類了三個 table 來管理規則，每個 table 則可以包含各自的 Chain，每條 Chain 則列了一條一條的 rule。

Table filter: 這個 Table 預設包含了三條規則鍊，INPUT，FORWARD，和 OUTPUT。INPUT 用來檢查要送達本機的封包，FORWARD 檢查經由本機轉送的

封包，OUTPUT 檢查本機產生發送出去的封包。要列出目前 filter table 的規則可以用 `iptables -L`。

Table nat: 包含三條鍊 PREROUTING，OUTPUT，POSTROUTING。當一個封包是要建立一條新的連線時候，就會交給 Nat Table 來處理。Table nat 最常見到的用途就是用來建立 NAT（Network Address Translation），也就是在封包進入這個 table 的時候就改變封包 Header 的 Source Address 來達到網路位址偽裝的效果。列出 table nat 的規則可以用 `iptables -t nat -L`。

Table mangle：包含了 PREROUTING，POSTROUTING，INPUT，OUTPUT，FORWARD 五條鍊。mangle 的本意是把東西剝碎，在這裡的意思是用來改變封包的內容，例如改變 TTL（Time To Live），或是 TOS（Type of Service）。列出 table mangle 的規則可以用 `iptables -t mangle -L`。

19.2 快速建立防火牆－Firestarter

安裝及啟動:Ubuntu 只要下指令 `apt-get install firestarter` 即會幫你安裝好 firestarter 了。並且也設定每次開機時候會自動載入預設或是你所定義的規則。安裝之後可以在程式集的系統程式裡面找到 Firestarter，如果執行時候不是以 root，則需要輸入 root 密碼才有設定防火牆的權限喔！

設定精靈:第一次執行的使用者，可以先從選單 Firewall->Run Wizard 來執行設定精靈。首先選擇用來對外連線的裝置，如果是區域網路就選擇 eth0，如果是使用 ADSL PPPoE 上網的話就要在 ADSL 連線的狀態下選擇 ppp0。



圖 19.2 選擇連線

接下來要選擇對內部的網路裝置，並且設定是否啟用 ICS (Internet Connection Sharing) 就是讓別台電腦可以透過這台電腦上網啦！若要啟用這項功能，你需要有兩張網路卡，路如對外連線用 eth0，對內就選擇 eth1，否則 firestarter 不會讓你繼續。



圖 19.3 分享網路連線設定

就這樣，最後把結果 Save 起來就可以了。防火牆啟動之後，就會在 Firestarter 的主畫面看到 Status 下面的箭頭圖示

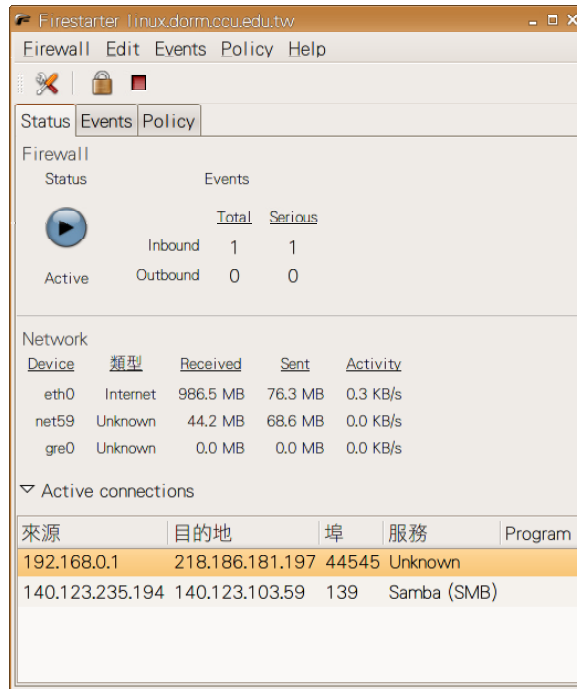
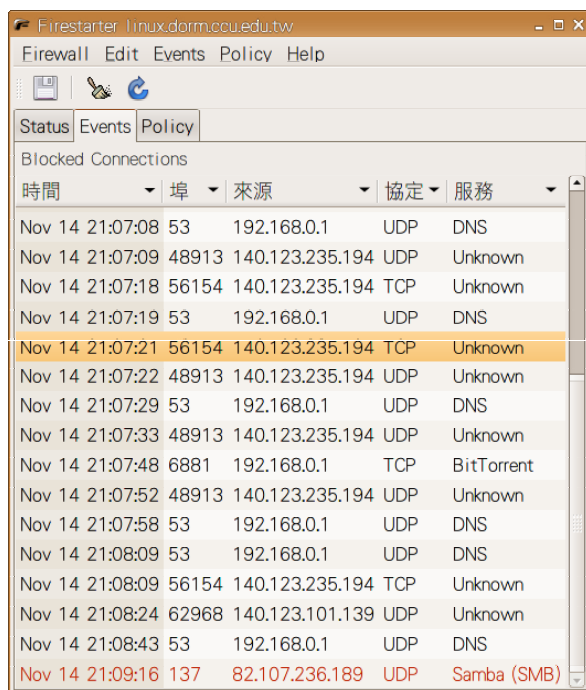


圖 19.4 Firestarter 啟動

要停止防火牆，回復最初的設定的話，就按工具列上的紅色 STOP 按鈕就行囉！

Firestarter 的使用介面分為三個 Tab

- **Status**：顯示目前防火牆的狀態，事件數目，各個裝置的流量，和目前的連線。
- **Events**：列出所有被擋下來的封包，你可以在各個事件上按右鍵，選擇同意從這個來源 IP 進來的連線（到本機所有的 port），或是同意所有連線到本機這個埠（port）的連線，或是同意這個來源 IP 連接到這個 port。



The screenshot shows the Firestarter application window with the 'Events' tab selected. The window title is 'Firestarter linux.dorm.ccu.edu.tw'. The menu bar includes 'Firewall', 'Edit', 'Events', 'Policy', and 'Help'. Below the menu bar is a toolbar with icons for home, mouse, and refresh. The main content area is titled 'Blocked Connections' and contains a table with the following columns: '時間' (Time), '埠' (Port), '來源' (Source), '協定' (Protocol), and '服務' (Service). The table lists various blocked connections, with the entry for Nov 14 21:07:21 highlighted in orange.

時間	埠	來源	協定	服務
Nov 14 21:07:08	53	192.168.0.1	UDP	DNS
Nov 14 21:07:09	48913	140.123.235.194	UDP	Unknown
Nov 14 21:07:18	56154	140.123.235.194	TCP	Unknown
Nov 14 21:07:19	53	192.168.0.1	UDP	DNS
Nov 14 21:07:21	56154	140.123.235.194	TCP	Unknown
Nov 14 21:07:22	48913	140.123.235.194	UDP	Unknown
Nov 14 21:07:29	53	192.168.0.1	UDP	DNS
Nov 14 21:07:33	48913	140.123.235.194	UDP	Unknown
Nov 14 21:07:48	6881	192.168.0.1	TCP	BitTorrent
Nov 14 21:07:52	48913	140.123.235.194	UDP	Unknown
Nov 14 21:07:58	53	192.168.0.1	UDP	DNS
Nov 14 21:08:09	53	192.168.0.1	UDP	DNS
Nov 14 21:08:09	56154	140.123.235.194	TCP	Unknown
Nov 14 21:08:24	62968	140.123.101.139	UDP	Unknown
Nov 14 21:08:43	53	192.168.0.1	UDP	DNS
Nov 14 21:09:16	137	82.107.236.189	UDP	Samba (SMB)

圖 19.5 Events Tab

- **Policy**：這裡讓你自訂規則，在 Events Tab 所作的更變也會顯示在這裡。
- **自訂規則**：限制流入流出的連線，可以用來限制外部存取內部的服務，或是限制內部的電腦對外的連線（例如不能用 BT、telnet 等等）。

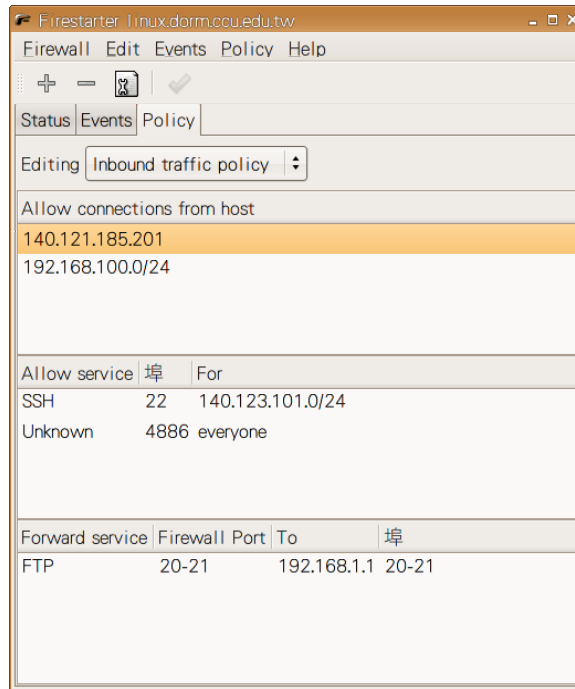


圖 19.6 限制進入連線

Firestarter 的設定值是阻擋任何進入本機的新連線，在這裡我們可以設定白名單，允許某些 IP/Network 連入。例如第一欄允許 140.121.185.201 這個 IP 和 192.168.100.開頭的所有 IP 連入本機。第二欄的限制更嚴格，除了限制連入的 IP，也限制了連接埠。在這裡我們允許 140.123.101.開頭的主機使用本機 port 22 的 ssh 服務，以及允許世界各地連到本機的 port 4886。第三欄是 Port-Forwarding 的設定，假使我們是一台 NAT 的閘道器，我們可以把外部連進來的連線轉接到內部的主機。

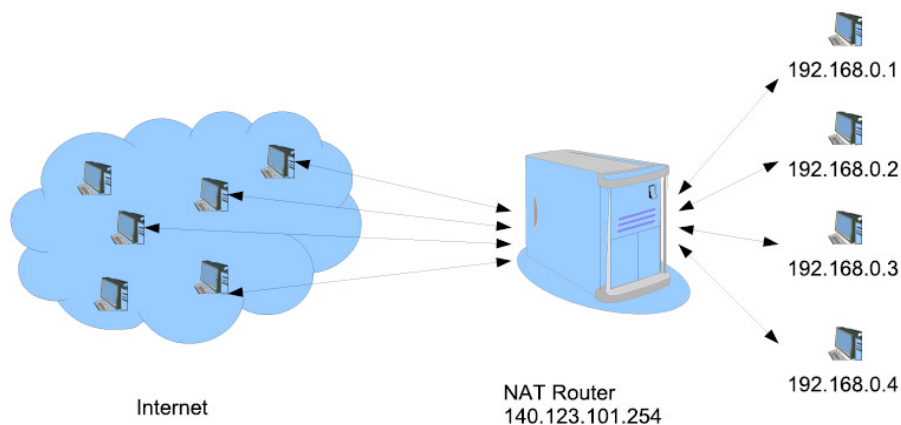


圖 19.7 NAT 示意圖

在外部看來只知道一台 14.1.23.5 的主機，但是這台 NAT Router 會把連線轉到內部各台對應的主機 10.0.0.1~10.0.0.5 上。這麼做的好處是，內部的主機只會收到 NAT Router 上面所設定的服務的連線，因此不用擔心會不小心有甚麼額外的服務遭到攻擊。例如假設 10.0.0.1 這台主機負責 http port 80 的服務，就算他上面也跑了其他的 service，也不會有外部的連線連的進來，因為會被 NAT Router 擋下，因此也不用擔心 http 以外的服務被攻擊。

限制對外連線

對外連線的限制分成兩種，第一種是允許所有連線，額外封鎖使用者自訂的黑名單。例如封鎖 MSN Messenger 的 port 1863 或是 ftp 的 21。

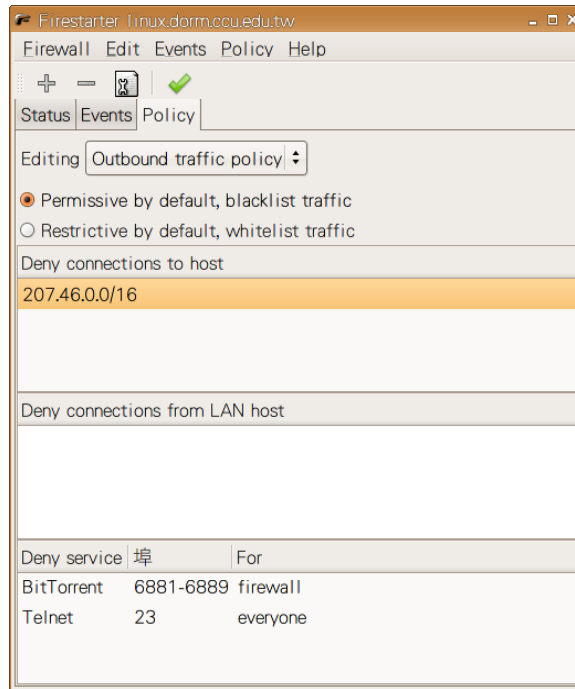


圖 19.8 黑名單範例

禁止連線到 207.48 開頭的網站，以及 BT、Telnet 服務，其他放行。

第二種的設定更嚴格，只允許使用者自訂的連線，其餘都封鎖。例如：只能連線到外部 http port 80。

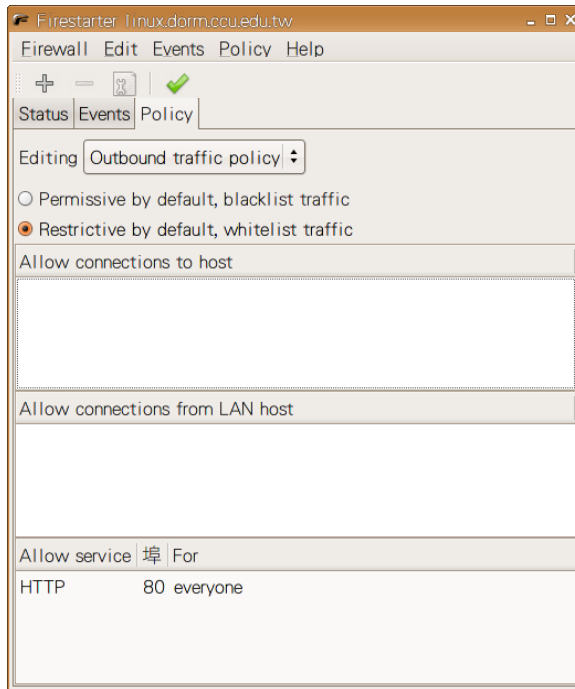


圖 19.9 白名單範例

只允許連線到 Http 服務。

CHAPTER 20

Beryl 3D Desktop

3D 桌面已經是未來的趨勢，近年來家用電腦普遍升級，而且不是家家戶戶都有一台，而是家家戶戶都有好幾台。一般人家裡大部份都是用 Windows，但是 Windows 大概也要好幾年才會出新的版本，相信大家都看膩了吧？雖然 Windows 的外觀是可以改的，不過能改的地方還是有限，不像 Linux 的 Window Manager 那麼的靈活自由，甚至像 jserv 大哥那樣自己寫了一個。聽人家說 2007 年，Windows 系列的作業系統也要推出新版本了，詳細的情況筆者沒去注意，不過不外乎應該是有 3D 的東西，以及透明度等等。當然囉～我們 Linux 這麼先進的作業系統，早在西元 2005 年就已經有了 3D 桌面。而這 3D 的 Window Manager 就是 Beryl。

20.1 Beryl 是什麼？

Beryl 是一個 Window Manager 兼 Composite Manager（讀者或許對此相當陌生，這也是近來 Xorg 發展提出的新概念，簡單來說，composite manager 允許對個別資源或細部的呈現作調整），利用 AIGLX 硬體繪圖加速技術透過 OpenGL 繪圖操作，畫出你想要的特殊效果呈現在螢幕上，如 3D 桌面、透明、還有抖動的效果等等。

支援 XGL/AIGLX 的 3D Window Manager & Composite Manager 是 compiz（現在也還存在），不過 compiz 是 Novell 在管的，社群裡有人對它不滿，所以後來社群版的分裂出來為 Beryl+XGL 的繪圖技術算是一個過渡期的方案，從 2006 年的 9 月底後，只要 upgrade 過後 compiz 就掛掉了，那是因為在此之後的 compiz 就分裂掉，變成了 Beryl，而且支援 AIGLX 這個嶄新的硬體繪圖加速技術，而 AIGLX 也正式整合到最近的 Xorg 開發。所以，Beryl 不需要如 compiz 一般依賴過渡性的 XGL 技術，雖然 Beryl 也能在 XGL 的環境下執行。XGL/AIGLX 是透過架構上的修改，才得以讓目眩神移的效果在 X Window System 上呈現。接下來就讓我們來看看 Beryl，它到底怎麼裝？有什麼特異功能呢？

20.2 安裝 Beryl

如果讀者用的是 ATI 顯示卡的話，那麼要先移除 xorg-driver-fglrx（註：這個驅動程式和內建的不一樣，在下面的 ATI 驅動程式有解釋）。因為目前 ATI 官方的驅動程式是 figlx，所以原廠的 driver 不支援 aiglx。但是要跑 aiglx，就要用在 xorg 7.1 裡的 opensource 版驅動程式 radeon，不過效能比較差。所以用 ATI 顯示卡的讀者們，要先用這個指令移除掉 ATI 官方的驅動程式。

```
$ sudo aptitude remove xorg-driver-fglrx
```

20.2.1 增加第三方套件庫來源

目前 Beryl 暫時還沒有被放到 edgy 的套件庫裡，所以我們要先在 /etc/apt/sources.list 加上套件庫來源，不過在我們開始這個步驟前，我們最好先更新一下桌面環境，指令步驟如下：

```
$ sudo apt-get install ubuntu-desktop
```

備註 依你的 Desktop Environment / Window Manager 而定，可以是 kubuntu-desktop，xubuntu-desktop。

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get dist-upgrade
```

接下來修改 `/etc/apt/sources.list`，來增加第三方的套件庫。建議你先備份好 `sources.list`。

```
$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
```

用 `gedit` 開啟 `sources.list`

```
$ sudo gedit /etc/apt/sources.list
```

加入以下來源

```
deb http://ubuntu.beryl-project.org/ edgy main
```

如果你是用 64 bit Ubuntu 的讀者，請換成這一行

```
deb http://compiz-mirror.lupine.me.uk/ edgy main-edgy main-edgy-amd64
```

下載並安裝新套件庫的 Apt Key。

```
$ sudo wget http://ubuntu.beryl-project.org/root@lupine.me.uk.gpg -O- | sudo apt-key add -
```

「-」的後面是大寫的 O（歐），不是 0（零）哦！

更新套件庫

```
$ sudo apt-get update
```

20.2.2 安裝顯示卡驅動程式

安裝 Nvidia 晶片顯示卡的驅動程式

你必需在 `/etc/apt/sources.list`，再加入下面這一行

```
deb http://seerofsouls.com/ edgy contrib
```

下載並安裝 Nvidia driver 套件庫的 Apt Key。

```
wget http://seerofsouls.com/keys/hawkwind.asc -O- | sudo apt-key add -
```

接著輸入

```
$ sudo apt-get update ; sudo apt-get install nvidia-glx ; sudo apt-get upgrade
```

這樣可以升級更新 `linux-restricted-modules`、`linux-restricted-modules-common` 這兩個套件。

然後讀者最好先備份一下 `/etc/X11/xorg.conf`

```
$ sudo nvidia-xconfig
```

這樣就會更新你的 `xorg.conf`，然後確定過你上面的步驟都做對了，新的驅動程式安裝後，請重開機。

ATI & Intel 晶片顯示卡的驅動程式

因為 ATI & Intel 的驅動程式有內建，原因是 Intel 的驅動程式，有完全開放原始碼，所以 Intel 的驅動程式有整合到 Xorg 裡。但是 ATI 並沒有開放原始碼，而且 ATI 官方非開放原始碼的驅動程式無法跑 AIGLX，不過也有人幫它寫 Open Source Driver，只是效能比較不好，但也有整合到 Xorg 裡，所以只要按照第一步，加上第三方套件庫後，安裝完 Beryl 後設定一下 `xorg.conf` 就可以用了。

20.2.3 安裝 Beryl

```
sudo apt-get install beryl emerald emerald-themes
```

因為相依性的關係，其他的 Beryl 相關套件會自動裝起來。

20.2.4 修改 `xorg.conf`

Nvidia 晶片顯示卡的設定

首先要修改 `/etc/X11/xorg.conf`。最好備份一下你的 `xorg.conf`。

```
$ sudo gedit /etc/X11/xorg.conf
```

加入下面粗體字的部份，來啟動 Beryl 的功能，要加在 Section "Screen" 裡，像是這個樣子：

```
...略...
    Modes      "1280x1024" "1152x864" "1024x768" "832x624" "800x600" ...略...
EndSubSection      #原本就有的
# Enable 32-bit ARGB GLX Visuals
Option "AddARGBGLXVisuals" "True"
EndSection      #原本就有的
```


還有 Section "Device" 要加上

```
Option          "TripleBuffer" "true"
```

像這樣

```
Section "Device"
    Identifier    "NVIDIA Corporation NV17 [GeForce4 MX 420]"
    Driver        "nvidia"
    Option        "TripleBuffer" "true"
EndSection
```

改完了以後就可以重開 X 啦！按下 `ctrl+alt+backspace`，然後重新登入，或是重新開機也可以。

ATI 晶片顯示卡的設定

修改 `/etc/X11/xorg.conf`。最好備份一下你的 `xorg.conf`。

```
$ sudo gedit /etc/X11/xorg.conf
```

Section "Device" 裡的 Drive，要改成 `"radeon"` 而不是 `"nvidia"`，Identifier 那一行不要改到哦（註：Identifier 那一行不一定跟筆者的一樣），像是這個樣子：

```
Section "Device"
    Identifier    "ATI Technologies, Inc. M9+ 5C63 [Radeon Mobility 9000 (AGP)]"
    Driver        "radeon"
    BusID        "PCI:0:16:0"
    Option        "UseFBDev" "true"
```

接著在這下面加入以下設定。

```
Option "DRI" "true"
Option "ColorTiling" "on"
Option "EnablePageFlip" "true"
Option "AccelMethod" "EXA"
Option "XAANoOffscreenPixmaps"
Option "RenderAccel" "true"
Option "AGPFastWrite" "on"
EndSection    #原本就有的
```

Intel 晶片顯示卡的設定

一樣修改 `/etc/X11/xorg.conf`。最好備份一下你的 `xorg.conf`。

```
$ sudo gedit /etc/X11/xorg.conf
```

在 Section "Device" 加上：

```
Option "XAANoOffscreenPixmaps" "true"
```

在 xorg.conf 的最下面加上：

```
Section "Extensions"
    Option "Composite" "true"
EndSection
```

20.2.5 啟動 Beryl X Window Manager

重新登入後開啟一個終端機，或是按下 Alt + F2 輸入 beryl-manager 或 beryl-start 就可以 work 了。接下就讓我們來看看，Beryl 到底有什麼好用的功能和熱鍵吧！當你的 Beryl work 後，第一個你可以看到的，就是這個啟動過程。

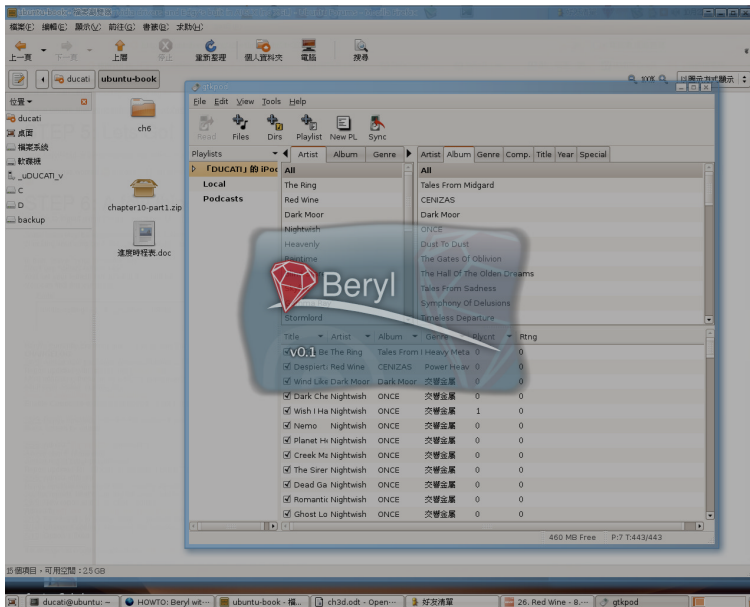


圖 20-1 啟動 Beryl

接著你可以在右上方，找到 Beryl Manager 像是一顆紅寶石，Beryl Manager 相當方便，可以讓你改變 Beryl 相關的設定，例如：Beryl 設定管理（可細部設定 Beryl，像是熱鍵等等的）、綠寶石主題管理器（可以用來設定你的視窗裝飾）、選擇 Window Manager 等功能，當你不想要啟動 Beryl 時，也可以換回原本的 Metacity，或是 KDE、XFCE 等等的 Window Manager。

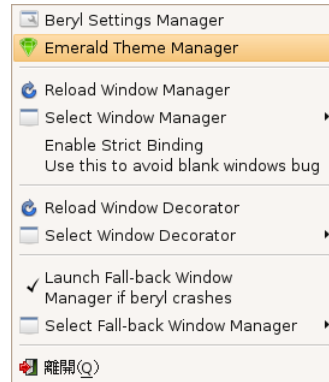


圖 20-2 Beryl Manager

Emerald Theme Manager 綠寶石主題管理器

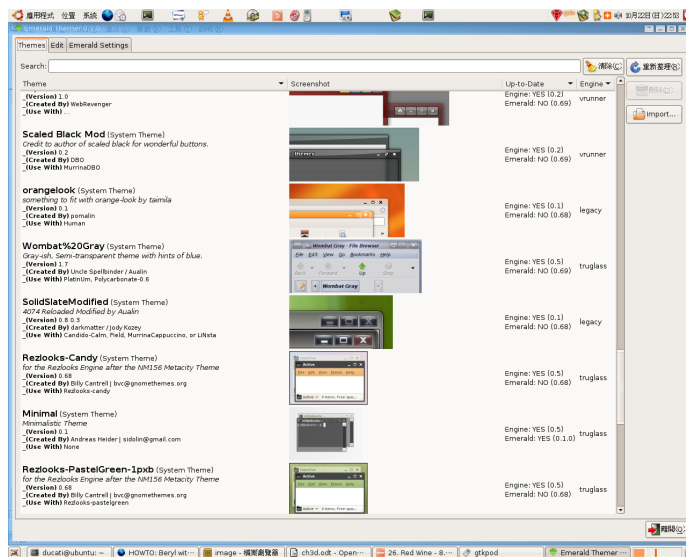


圖 20-3 Emerald Themer

雖然內建就大約有二十幾種的 theme 可以選擇，不過讀者也可以上網找別的主题來安裝，網址<http://www.gnome-look.org/>。

Beryl HotKeys 與功能簡介

1. 切換視窗 Alt + Tab

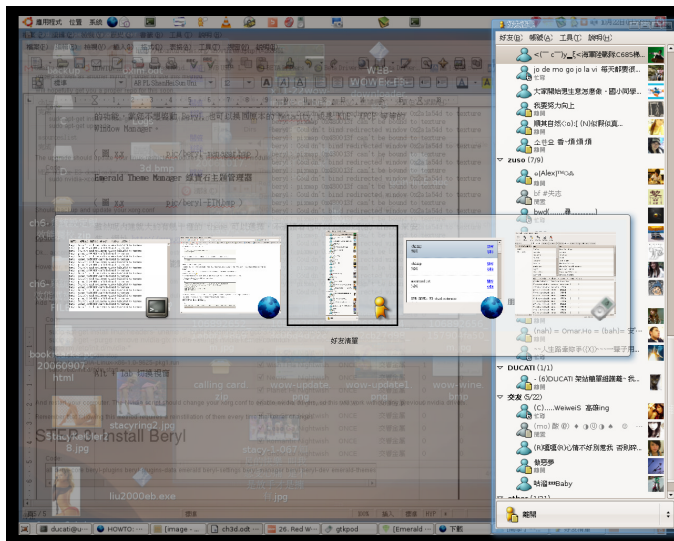


圖 20-4 切換視窗時，還會產生縮圖方便預覽。

2. 果凍 QQ 的效果，拉動視窗即可。或是按住 Alt 再抓住視窗任一位置，抓不到最上面的邊框時可以用這招。

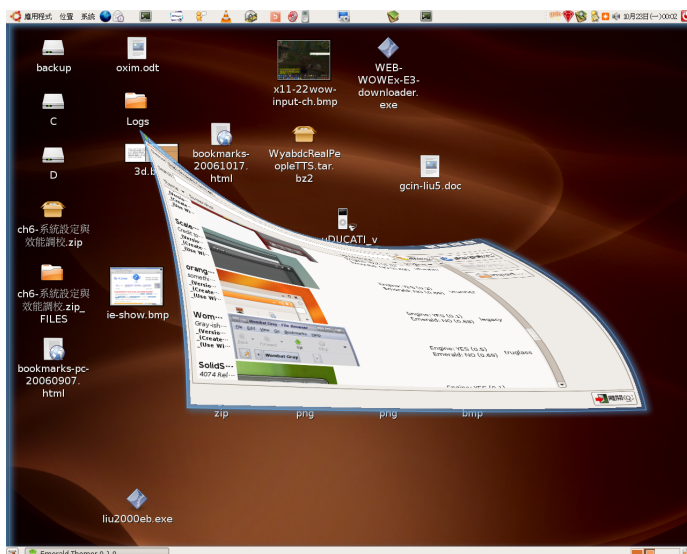


圖 20-5 果凍 QQ 的效果

3. 切換桌面，Ctrl+Alt+方向鍵（上下左右都可以），或是將指標移到桌面的左或右邊，滾動滑鼠滾輪也可以。



圖 20-6 切換桌面

4. 自由旋轉，先按住 Ctrl+Alt 然後再點滑鼠左鍵，就可以自由旋轉你的桌面囉！

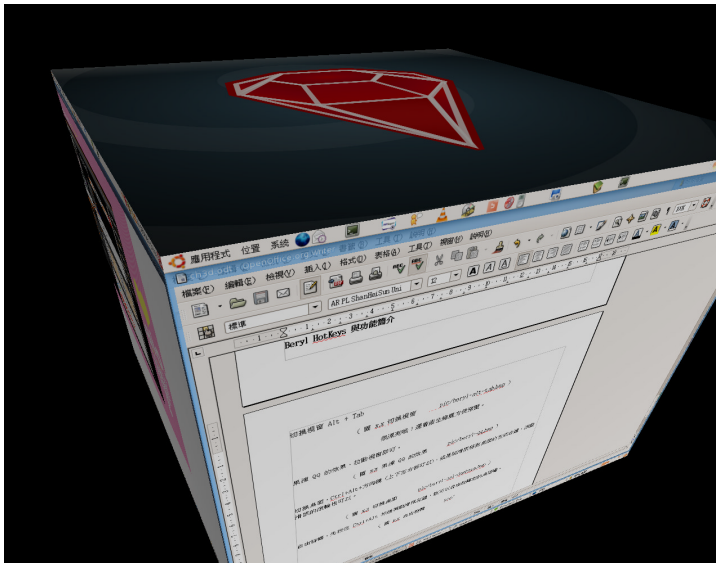


圖 20-7 自由旋轉

7. 所有桌面上的視窗都排列整齊，按下 F8。和指標移到左下方功能一樣。

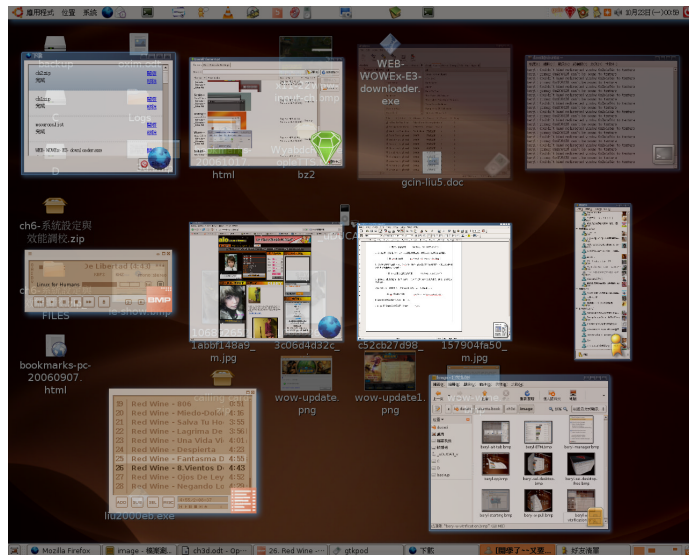


圖 20-10 所有桌面上的視窗都排列整齊

8. 這個桌面上的視窗都排列整齊，按下 F9。和指標移到右上方功能一樣。按住 Alt 再用滑鼠左鍵，就可以抓著視窗的任何一個地方。



圖 20-11 掀開視窗

9. 攤開所有桌面，Ctrl + Alt + PageDown。

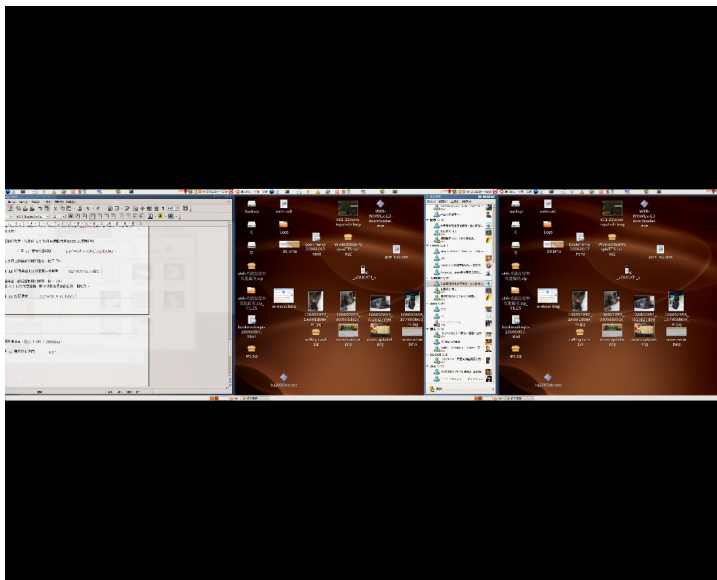


圖 20-12 攤開所有桌面

10. 全部視窗縮小，Ctrl + Alt + D。

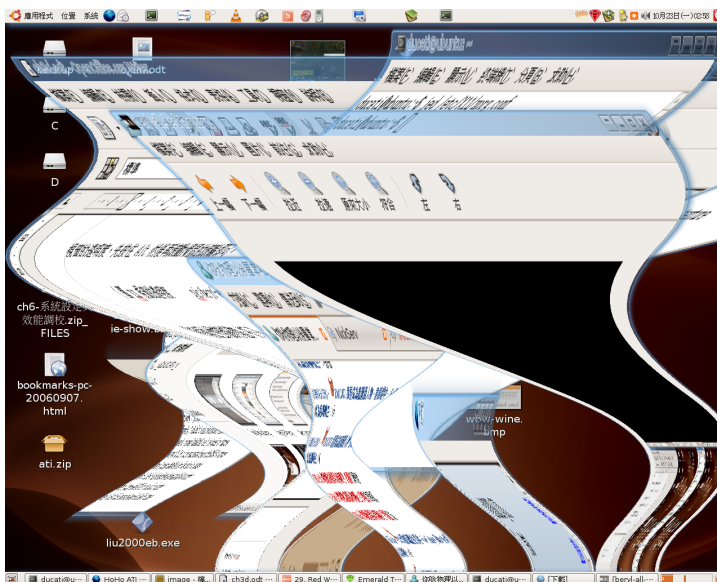


圖 20-13 全部視窗縮小

11. F8 後再全部視窗縮小。

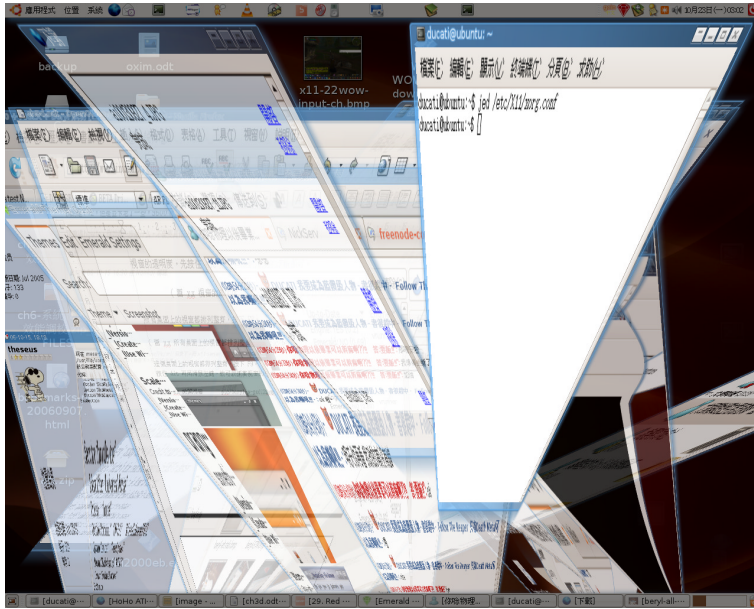


圖 20-14 F8 後再全部視窗縮小

12. Ctrl + Alt + H 寬度全滿，再按一次變回來。
13. Ctrl + Alt + V 高度全滿，再按一次變回來。
14. Ctrl + Alt + S 視窗捲起，再按一次變回來。
15. 讓 Beryl 變成慢動作，按下 Shift + F10 可以讓 Bery 的效果變慢，再按一次即可恢復。

20.2.6 將 Beryl 加到作業階段

這樣就可以一進到 X 就有 Beryl，而不用先進去 X 再開 Beryl。方法：選擇系統 --> 偏好設定 --> 作業階段 --> 初使啟動程式 --> 新增，加上底下這一行：

```
/usr/bin/beryl-manager
```

可是要記得哦！Beryl 和 DirectX 的遊戲是不能一起執行的。例如：WoW、Steam 等。那麼剩下的就是依讀者的喜好，調整 Beryl 的設定值，其實還有很多特效可以用，也可以將 Beryl 調整的更實用。

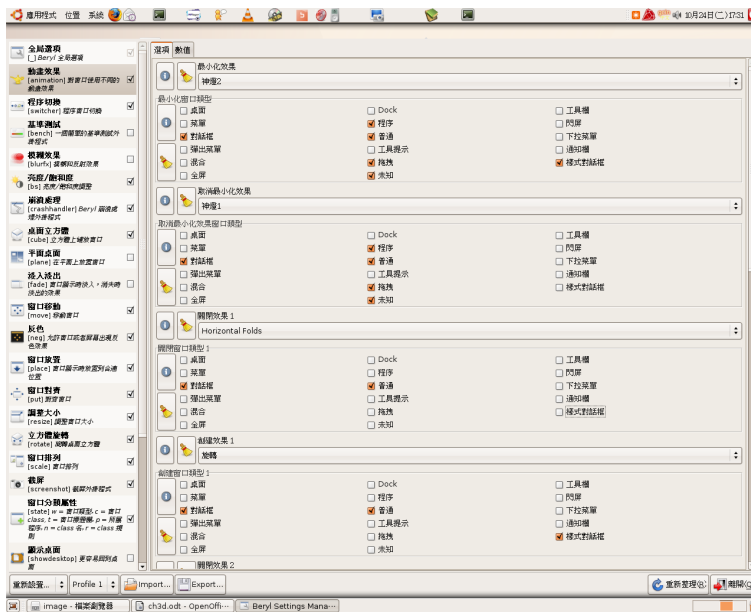


圖 20-15 Beryl Settings Manager

CHAPTER 21

Edgy 新增加的程式

21.1 Tomboy 好用的筆記本工具

Tomboy 是這次新增的筆記工具，比以往的筆記工具多了一些新意，可以隨時將想到的點子記下來之外，也可以隨時做改變，你可以為筆記加上粗體、斜體、刪除線，重要的是！這個程式可以隨時將你的筆記轉成 HTML 的檔案，也可以直接從印表機上印出來，或是從舊的便條裡匯入資料。

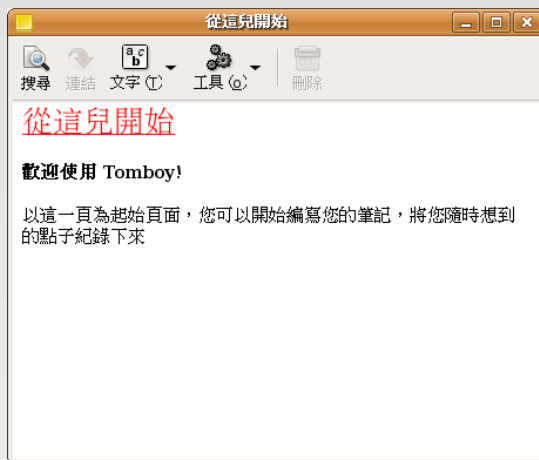



圖 21.1 Tomboy

 安裝方式

```
sudo apt-get install tomboy
```

21.2 F-Spot Photo Manager 相簿管理程式

F-Spot 是採用匯入的方式管理，也就是說你可以將想加入相簿的圖片匯入來做管理，當然也可以自動偵測你所接上去的數位像機。匯入後你可以直接在 F-Spot 修改圖片，或是當成 Show 圖片程式來用，你也可以設定標籤，全螢幕、幻燈片等等的都是基本的功能。F-Spot 主要是以日期來做圖片的分類，這樣可以很好分辨是什麼時候產生的圖，而不用去理會檔名。這樣看來真的是一個蠻方便的相簿管理程式。

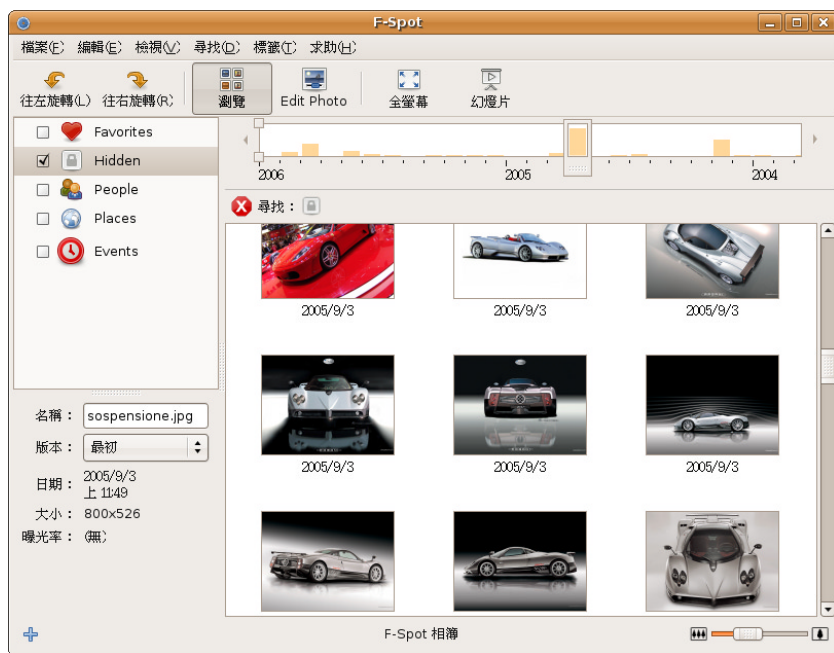


圖 21.2 F-Spot

EB440

次世代Linux-Ubuntu玩全手冊

親愛的讀者：

感謝您購買上奇科技出版的圖書，為提供給您更好的服務，並且幫助我們更加瞭解您的需求。

請您填妥下列資料，將讀者回函卡剪下直接寄回（免貼郵票）或是傳真至本公司，您的建議及鼓勵將是我們成長的動力。

姓名：_____ 性別：_____ 出生日期：_____年_____月_____日
公司名稱：_____ 部門：_____ 行動電話：_____
電話：_____ E-Mail：_____
地址：_____市/縣_____鄉/鎮/區_____路/街_____段_____巷_____弄_____號_____樓

讀者意見

Q1. 我的職業是

學生 上班族 SOHO族 其他

Q2: Linux的使用經驗

剛開始學習 1年以上

Q3. 我買這本書是因為...? (可複選)

內容充實及易讀 作者 編排及封面 出版社 印刷品質精美
 他人推薦或老師指定 書本的附加價值 (附贈之光碟) 書中有別本書沒講到的東西

Q4. 您對本書內容感到

查閱清楚 解說清楚 圖例精美 其他_____

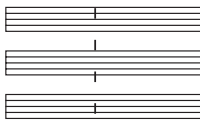
Q5. 我也有興趣購買以下主題的書

網路安全 系統管理 Windows Server 其他_____

Q6. 本書需要改進的地方？

現在就傳真或寄回此函 傳真熱線：02-8792-3002
上奇網站：www.grandtech.com.tw





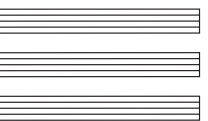
廣告回函
台灣北區郵政管理局登記證
第77699號

寄件人：

上奇科技股份有限公司
出版事業部
收

1114-53

台北市內湖區瑞光路76巷33號3樓



上奇科技

讀者回函卡



1. 書籍或光碟檔案遺漏
請至上奇出版網站
(<http://www.grandtech.com.tw/book/index.asp>)
的勘誤表區下載

2. 書附光碟毀損(光碟不需支付費用)
煩請附回郵信封(掛號30元/平信15元)
寄至：台北市內湖區114瑞光路76巷33號3樓 出版部
並註明索取書名與作者名稱之光碟及聯絡姓名、住址、電話

3. 書籍缺損
請來信 (editor@grandtech.com.tw) 註明缺損書籍的狀況，
並留下您的聯絡姓名、電話、住址，我們將有專人與您聯繫。

如還想知道更多的訊息與完整的解決方案，
來上奇科技網站就對了！
<http://www.grandtech.com.tw>
如有任何問題，歡迎來信editor@grandtech.com.tw