Debian 參考手册

Osamu Aoki <osamu\#at\#debian.org> Asho Yeh <asho\#at\#debian.org.tw> "作者" 235

CVS, 週四一月 18 11:53:38 UTC 2007

#### 摘要

Debian 參考手冊 (http://qref.sourceforge.net/) 的目標在為 Debian 的使用者提供全面性的指導,就如同一本安裝之後的使用手冊。本文中以大量的 shell 指令範例來涵蓋系統管理的各個面向。包括的主題有: Debian 系統的基本概念、系統安裝提示、 Debian 套件管理、 Debian 中的 Linux 核心、系統微調、 gateway 建立、文字編輯器、 CVS 、程式設計、並為非 開發者說明 GnuPG 等,每個主題都包括基礎指南、實用技巧和其它相關的資訊。

# 版權聲明

Copyright (c) 2001–2005 by Osamu Aoki <osamu#at#debian.org>. Copyright (Chapter 2) (c) 1996–2001 by Software in the Public Interest.

This document may be used under the terms of the GNU General Public License version 2 or higher. (http://www.gnu.org/copyleft/gpl.html)

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions, except that this permission notice may be included in translations approved by the Free Software Foundation instead of in the original English.

# Contents

1	序言	•		1
	1.1	官方文	件	1
	1.2	文件慣	例	1
	1.3	scripts	;範例	2
	1.4	基本設	定	2
	1.5	Debiar	n distributions 的基本概念	2
2	Deb	ian 基码	楚	5
	2.1	Debiar	archives	5
		2.1.1	目錄結構	5
		2.1.2	Debian distributions	6
		2.1.3	stable 發行版本	6
		2.1.4	The testing distribution	6
		2.1.5	The unstable distribution	7
		2.1.6	The frozen distribution	7
		2.1.7	Debian distribution 開發代號 (codenames)	7
		2.1.8	已用過的開發代號	8
		2.1.9	開發代號的來源	8
		2.1.10	pool 目錄	8
		2.1.11	sid 的歷史記錄	8
		2.1.12	在 incoming 中的上傳套件	9
		2.1.13	找回舊套件	9
		2.1.14	Architecture sections	9
		2.1.15	源碼	0

2.2	<b>Debian</b> 套件管理系統		
	2.2.1	Debian 套件概觀	10
	2.2.2	Debian 套件格式	11
	2.2.3	Debian 套件檔名的命名	11
	2.2.4	保留本機設定檔	11
	2.2.5	Debian maintenance scripts	12
	2.2.6	套件的優先等級	12
	2.2.7	虚擬套件	13
	2.2.8	套件相依性	13
	2.2.9	何爲"pre-depends"	14
	2.2.10	套件狀態	14
	2.2.11	阻止套件升級	14
	2.2.12	原始碼套件	15
	2.2.13	編譯原始碼套件	15
	2.2.14	新建Debian套件	16
2.3	Debia	n系統升級	16
	2.3.1	dpkg	16
	2.3.2	APT	16
	2.3.3	dselect	17
	2.3.4	在運行中的系統升級	17
	2.3.5	下載和快取.deb檔案	17
	2.3.6	升級記錄	17
2.4	Debia	n 開機程序	17
	2.4.1	init 程式	17
	2.4.2	Runlevels	18
	2.4.3	自訂 RunLevel	18
2.5	多樣性	±支援	19
2.6	國際化	• • • • • • • • • • • • • • • • • • • •	19
2.7	Debia	n和系統核心	19
	2.7.1	編譯非 Debian 核心的原碼	19
	2.7.2	自定核心建立工具	20
	2.7.3	模組處理措施	20
	2.7.4	移除舊版核心套件	20

3	Deb	bian 系統安裝提示 23			
	3.1	一般Linux安裝提示	23		
		3.1.1 硬體相容性	23		
		3.1.2 確定 PC 硬體和晶片組	24		
		3.1.3 在 Debian 下確定 PC 硬體	24		
		3.1.4 在其它 OS 下確定 PC 硬體	25		
		3.1.5 Lilo 迷思	25		
		3.1.6 GRUB	25		
		3.1.7 選擇開機軟碟	26		
		3.1.8 安裝系統	26		
		3.1.9 網路所需的主機名和IP地址	27		
		3.1.10 使用者帳號	27		
		3.1.11 建立檔案系統	28		
		3.1.12 DRAM記憶體指導	31		
		3.1.13 Swap空間	31		
	3.2	Bash 設定	31		
	3.3	滑鼠設定	32		
		3.3.1 PS/2滑鼠	32		
		3.3.2 USB 滑鼠	34		
		3.3.3 觸控版(Touchpad)	34		
	3.4	NFS設定	35		
	3.5	Samba設定	35		
	3.6	印表機設定	36		
		3.6.1 lpr/lpd	36		
		3.6.2 CUPS <sup>TM</sup>	37		
	3.7	桌上型電腦使用CRON	37		
	3.8	其它主機安裝提示	38		
		3.8.1 初始化安装完成後再装些什麼	38		
		3.8.2 模組	38		
		3.8.3 CD-RW基本步驟	38		
		3.8.4 大記憶體和關機自動斷電	38		

		3.8.5	無法存取某此站台的怪問題	39
		3.8.6	PPP 撥號設定	40
		3.8.7	/etc/中的其它設定文件	40
4	Deb	ian指南	ġ	41
	4.1	準備開	始	41
		4.1.1	用 root 登入 shell 提示符號	41
		4.1.2	針對新手設定的最精簡環境	42
		4.1.3	增加帳號	42
		4.1.4	切换虚凝主控台	43
		4.1.5	如何關機	43
		4.1.6	把玩時間	43
	4.2	Midni	ght Commander (MC)	44
		4.2.1	加強您的 MC	44
		4.2.2	啓動 MC	44
		4.2.3	MC 的檔案管理功能	44
		4.2.4	MC 的命令列技巧	45
		4.2.5	MC 的编輯器	45
		4.2.6	MC 的檢視器	46
		4.2.7	MC 自動啓動特性	46
		4.2.8	MC 的FTP虛擬檔案系統	46
	4.3	Unix-l	ike 工作環境	46
		4.3.1	特殊的組合鍵	47
		4.3.2	基本的 Unix 指令	47
		4.3.3	The command execution	50
		4.3.4	簡單的命令	51
		4.3.5	指令執行和環境變數	51
		4.3.6	指令收尋路徑	51
		4.3.7	命令列參數	52
		4.3.8	Shell wildcards	52
		4.3.9	指令的傳回值	53

		4.3.10 標準的指令順序         53
		4.3.11 指令別名
	4.4	Unix-like 文字處理
		4.4.1 正規表示法
		4.4.2 置换表示法
	4.5	Unix-like 檔案系統 57
		4.5.1 Unix 檔案基礎 57
		4.5.2 Debian 的檔案系統 58
		4.5.3 檔案和目錄的存取權限 59
		4.5.4 時間戳記
		4.5.5 連結
		4.5.6 Named pipes (FIFOs) 62
		4.5.7 Sockets
		<b>4.5.8</b> Device files
		4.5.9 /proc 檔案系統 64
	4.6	X Window 系統
		4.6.1 啓動 X Window System 64
		4.6.2 X Window 系統的選單功能
		4.6.3 X Window 系統的鍵盤順序 64
	4.7	後續的學習指南
5	將 d	listribution 升級至 stable, testing 或 unstable 67
	5.1	從 Potato 到 Woody
	5.2	升级前的準備
	5.3	升級
		5.3.1 使用 dselect
		5.3.2 使用 apt-get 69
6	Dah	ian 春件管理系统 71
U	6 1	
	0.1	1     ····································
		0.1.1 工文的去口旨在一六
		- <b>0.1.</b> <i>L</i> - 刀仪则上共 - · · · · · · · · · · · · · · · · · ·

6.2	體驗 [	Pebian 套件管理	72
	6.2.1	設定 APT	72
	6.2.2	tasks 安裝	72
	6.2.3	aptitude	73
	6.2.4	dselect	74
	6.2.5	使用 APT 來維持 Debian 發行版本	74
	6.2.6	aptitude,apt-get和apt-cache命令	75
6.3	Debia	n 生存指令	76
	6.3.1	檢查 Debian 中的 bugs 並尋求幫助	76
	6.3.2	APT 升级的錯誤排除方法	77
	6.3.3	使用 dpkg 來救援	77
	6.3.4	回復套件的選取狀態的資料	78
	6.3.5	在 /var 崩潰後救援系統	78
	6.3.6	把套件安装到一個無法開機的系統	78
	6.3.7	如果 dpkg 指令壞了怎麼辦	79
6.4	Debia	n 神技之指令	79
	6.4.1	檔案中的資訊	79
	6.4.2	套件的資訊	80
	6.4.3	Unattended installation with APT	80
	6.4.4	重新設定已安裝的套件	81
	6.4.5	移除和清除套件	81
	6.4.6	維持舊的套件	82
	6.4.7	stable/testing/unstable 混合系統	82
	6.4.8	删除暫存的套件檔	82
	6.4.9	記錄/複製系統設定	83
	6.4.10	把套件引入 stable 系統	83
	6.4.11	本地端的套件 archive	84
	6.4.12	轉換或安裝外來的二進位套件	85
	6.4.13	Automatically install command	85
	6.4.14	驗證已安裝的套件檔	85
	6.4.15	sources.list 最佳化	86

	6.5	Debia	n 其它特別之處	. 86
		6.5.1	dpkg-divert 指令	. 86
		6.5.2	equivs 套件	. 86
		6.5.3	Alternative 指令	. 86
		6.5.4	Runlevel	. 87
		6.5.5	停止 daemon 服務程式	. 87
7	Deb	ian下的	的Linux核心	89
	7.1	核心(重	重新)编譯	. 89
		7.1.1	核心標頭檔	. 89
	7.2	透過p	proc 檔案系統調整核心	. 90
		7.2.1	開啓太多檔案	. 90
		7.2.2	清除磁碟緩衝區時間	. 90
		7.2.3	慢速且記憶體小的機器	. 90
	7.3	核心2	.6 种 udev	. 91
8	Deb	ian技巧	5	93
	8.1	啓動系	统	. 93
		8.1.1	「我忘了root密碼!」(1)	. 93
		8.1.2	「我忘了root密碼!」(2)	. 94
		8.1.3	無法啓動系統	. 94
		8.1.4	「我不想直接啓動到 X!」	. 95
		8.1.5	其它用於啓動提示的技巧	. 95
		8.1.6	設定 GRUB 啓動參數	. 96
	8.2	活動記	2錄	. 96
		8.2.1	記錄shell活動	. 96
		8.2.2	記錄 X 操作	. 97
	8.3	拷貝及	建立子目錄	. 97
		8.3.1	拷貝整個子目錄的基本指令	. 97
		8.3.2	ср	. 98
		8.3.3	- tar	. 98
		8.3.4	pax	. 98

	8.3.5	cpio
	8.3.6	afio
8.4	差異備	·份與資料同步
	8.4.1	使用rdiff進行差異備份 100
	8.4.2	使用pdumpfs進行每日備份 100
	8.4.3	使用 RCS 進行定期差異備份100
8.5	系統凍	
	8.5.1	中止一個進程
	8.5.2	Alt-SysRq
8.6	記住這	些可爱的小指令
	8.6.1	Pager
	8.6.2	釋放記憶體
	8.6.3	設定時間(BIOS) 102
	8.6.4	設定時間(NTP)
	8.6.5	如何控制 console 的特色,例如螢幕保護程式
	8.6.6	查詢系統管理資料程式庫103
	8.6.7	禁用聲音(響鈴)
	8.6.8	控制台 (console) 上的錯誤訊息
	8.6.9	正確設定控制台 (console) 檔案類型104
	8.6.10	恢復 console 到穩定狀態
	8.6.11	將 DOS 下的文字檔案轉換為Unix檔案類型
	8.6.12	利用 recode 來轉換文字檔104
	8.6.13	Regular-expression substitution
	8.6.14	使用 script 來編輯檔案
	8.6.15	提取來源檔案修改部分合併到更新包106
	8.6.16	分割大檔案
	8.6.17	從文字檔格式的表格中抽取資料107
	8.6.18	精巧的管道指令辅助script 108
	8.6.19	查看每個檔案的輕巧 script
	8.6.20	輕巧的 Perl Script
	8.6.21	從網頁上取得文件或通信清單(mailing list)

		8.6.22	列印網頁
		8.6.23	列印幫助頁面
		8.6.24	合併兩個Postscript或PDF檔案111
		8.6.25	指令耗時
		8.6.26	nice指令
		8.6.27	日程安排 (cron, at)
		8.6.28	用screen來切換 console
		8.6.29	網路測試基礎
		8.6.30	清理(flush)本地端的 Mail 佇列114
		8.6.31	删除已凍結的信件
		8.6.32	重新寄送mbox中的信件114
		8.6.33	清理檔案內容
		8.6.34	空檔案
		8.6.35	chroot
		8.6.36	怎樣檢查 hard links117
		8.6.37	mount硬碟上的映象檔案 117
		8.6.38	Samba
		8.6.39	外來檔案系統的操作工具118
	8.7	打錯字	的情况
		8.7.1	rm -rf .*
		8.7.2	rm /etc/passwd
0	Dob	ion \$4	a 润敷 110
9	0 1	初始化	系统的提示 119 119
	9.1	011	自定 init 命令稳 110
		9.1.1	日又 IIII 中マ禍 · · · · · · · · · · · · · · · · · · ·
		9.1.2	硬體量佳化(Ontimizing hardware) 119
	92	方1.5 左取限	制(Restricting access)
	1.2	971	A 用 PAM 來控制容λ程序 121
		9.2.1	"食何 CNII au 不支援 whool 群组" 127
		9.2.2	Purposes of standard groups
		9.2.3	

9.2.4	更安全的工作環境-sudo12	22
9.2.5	服務的存取限制	<u>2</u> 3
9.2.6	集中式的認證功能 – LDAP	<u>2</u> 3
燒錄器		24
9.3.1	簡介	24
9.3.2	<u>方法一:modules</u> +lilo12	24
9.3.3	方法二: recompile the kernel	25
9.3.4	設定步驟	25
9.3.5	光碟映像檔 (可開機)	26
9.3.6	寫錄到燒錄機上(R, RW):12	26
9.3.7	產生一個光碟映像檔12	27
9.3.8	Debian 光碟映像檔	27
9.3.9	備份您的資料到 CD-R 12	27
9.3.10	拷貝音樂光碟	28
9.3.11	燒錄 DVD-R,DVD-RW 和 DVD+RW	28
<b>X</b>		28
9.4.1	X 套件	<u>2</u> 9
9.4.2	X 的硬體偵測	30
9.4.3	X 伺服器	30
9.4.4	X clients	32
9.4.5	X sessions	33
9.4.6	在TCP/IP的環境中使用X Window	36
9.4.7	連接到遠端的X server - xhost1	36
9.4.8	連線到遠端的 X server – ssh 13	37
9.4.9	X 終端機模擬器 – xterm13	37
9.4.10	X 資源	37
9.4.11	X的按鍵和指標按鈕對應13	38
9.4.12	在 X 裡取得 root	38
9.4.13	在 X 的 TrueType 字型	<b>1</b> 0
9.4.14	在 X 中的網頁瀏覽器 14	<del>1</del> 2
SSH .		12
	9.2.49.2.59.2.6烧錄器9.3.19.3.29.3.39.3.49.3.59.3.69.3.79.3.89.3.99.3.109.3.11X9.4.19.4.29.4.39.4.49.4.59.4.69.4.79.4.89.4.99.4.109.4.119.4.129.4.139.4.145.145.14	92.4       更安全的工作環境-sudo       12         92.5       服務的存取限制       12         92.6       集中式的認證功能-LDAP       12         9.3.1       簡介       12         9.3.2       方法一:modules + 1110       12         9.3.3       方法二:recompile the kernel       12         9.3.4       設定步驟       12         9.3.5       光碟映像檔 (可閒機)       12         9.3.6       寫錄到逸線也(R, RW):       12         9.3.7       產生一個光碟映像檔       12         9.3.8       Debian 光碟映像檔       12         9.3.9       備份您的資料到 CD-R       12         9.3.10       拷員音樂光碟       12         9.3.11       兢兢 DVD-R, DVD-RW 和 DVD+RW       12         9.3.11       兢兢 DVD-R, DVD-RW 和 DVD+RW       12         9.4.1       X 条件       12         9.4.3       X 伺服器       13         9.4.4       X clients       13         9.4.5       X sessions       15         9.4.6       & TCP/IP的環境中使用X Window       13         9.4.7       建填到達場的X server - sh       15         9.4.8       違爆到達場的X server - sh       15         9.4.1       X 約束過       14         9.4.10       X

		9.5.1	SSH 的基礎
		9.5.2	Port forwarding for SMTP/POP3 tunneling
		9.5.3	以少許密碼連線 – RSA
		9.5.4	與不同的 SSH 用户端程式溝通145
		9.5.5	設定 ssh-agent
		9.5.6	SSH 錯誤解決
	9.6	Mail .	
		9.6.1	郵件傳送代理 (MTAs)146
		9.6.2	撷取信件l-Fetchmail149
		9.6.3	處理郵件 – Procmail
		9.6.4	使用 crm114 來處理垃圾郵件 149
		9.6.5	閱讀郵件 – Mutt149
	9.7	Localiz	$      zation (110n)  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $
		9.7.1	localization 的基礎
		9.7.2	Locales
		9.7.3	locales 簡介
		9.7.4	啓動 locale 支援
		9.7.5	啓動特定的 locale
		9.7.6	ISO 8601 日期格式 locale
		9.7.7	US (ISO-8859-1) 的範例
		9.7.8	France with Euro sign (ISO-8859-15)
		9.7.9	多語系的 X 視窗系統範例154
		9.7.10	其他的 X 輸入法
		9.7.11	X 終端機
		9.7.12	支援 UTF-8 的 X 終端機158
		9.7.13	支援 UTF-8 的 framebuffer 主控台
		9.7.14	locales 之外
	9.8	Multili	ingualization (m17n)
10	細叶	扒户	
10	網路	· 武疋	161 • # -#
	10.1	IP 網路	▶ 埜ç

10.2 低階網路設定
10.2.1 低階網路設定 – 使用 ifconfig 和 route
10.2.2 低階網路設定—ip164
10.2.3 設定 Wi-Fi 裝置
10.2.4 設定 PPP 裝置
10.3 對主機命名
10.3.1 主機名稱
10.3.2 郵件名稱
10.4 網域名稱服務 (DNS)
10.4.1 resolver
10.4.2 管理名稱伺服器的資訊 – resolvconf
10.4.3 快取已查閱的名稱—nscd, dnsmasq, pdnsd, bind9
10.4.4 提供網路名稱服務-bind 170
10.5 安裝網路界面使用 DHCP171
10.6 Debian 的高階網路設定
10.6.1 高階網路設定-使用ifupdown
10.6.2 高階網路設定-ifupdow邏輯裝置定義
10.6.3 神奇的網路設定方式
10.6.4 利用 laptop-net 自動設定網路
10.6.5 利用network-manager自動設定網路
10.7 處理核心的非一致性裝置命名規則
10.8 觸發(Triggering)網路設定178
10.8.1 在開機時觸發(trigging)網路設定
10.8.2 觸發網路設定 – 使用hotplug
10.8.3 觸發網路設定 – 使用ifplugd
10.8.4 觸發網路設定—使用 waproamd
10.8.5 網路設定和 PCMCIA
10.9 多階段(Multi-stage)對應
10.10網路服務設定
10.11解決網路問題
10.12建立通訊閘路由器
10.12.1 Netfilter 設定
10.12.2 管理多重的網路連線

11	編輯	器		189
	11.1	流行的約	编輯器	189
	11.2	應急的約	编輯器	189
	11.3	Emacs <sup>5</sup>	种 Vim	190
		11.3.1	Vim 指示	190
		11.3.2	Emacs 提示	190
		11.3.3	啓動編輯器	190
		11.3.4	Editor 指令摘要 (Emacs , Vim) ......................	191
		11.3.5	Vim 設定	193
		11.3.6	Ctags	193
		11.3.7	將高亮度顯示的螢幕內容轉爲 HTML 文件	193
		11.3.8	用 vim 分割螢幕	193
10	出大	抗制灭的	-	105
12	101	AT WAS	rent Versions System (CVS)	195
	12.1	12 1 1	cent versions System (CVS)	195
		12.1.1	文永(V3)内心語 · · · · · · · · · · · · · · · · · · ·	195
		12.1.2	CVS Session 乾內	193
		12.1.5	CVS 带无问题所厌为么	199
	10.0	12.1.4	CV3 # 7	200
	12.2	12 2 1	stoft	200
		12.2.1	文衣 Subversion 网版語	200
		12.2.2	水 CVS 福米平将沙玛 Subversion	201
		12.2.3		201
13	程式	設計		203
	13.1	從哪兒	開始	203
	13.2	Shell .		203
		13.2.1	Bash – GNU 標準互動式 shell	203
		13.2.2	POSIX shells	204
		13.2.3	Shell 參數	205
		13.2.4	Shell 重導 (redirection)	205
		13.2.5	Shell 條件式	206

		13.2.6 命令列操作	207
	13.3	Awk	207
	13.4	Perl	209
	13.5	Python	210
	13.6	Make	211
	13.7	C	212
		13.7.1 簡單的 C 程式 (gcc)	213
		13.7.2 除錯	213
		13.7.3 Flex – 更好的 Lex	215
		13.7.4 Bison – 更好的 Yacc	215
		13.7.5 Autoconf	216
	13.8	Web	217
	13.9	文件準備	217
		13.9.1 roff 排版	217
		13.9.2 SGML	218
		13.9.3 TeX/LaTeX	219
		13.9.4 文學編程	220
	13.10	)包裝套件	221
		13.10.1 包裝一個單一二進位套件	221
		13.10.2 使用工具來進行包裝工作	222
11	Cou	PC	222
14	1/1 1	立 安特 Chui DC	223
	14.1	使用 Creat DC	223
	14.2	资理 Chure Ch	223
	14.5	下 Glur G · · · · · · · · · · · · · · · · · ·	224
	14.4	在您用在式+ 夜州 Ghur G	220
		14.4.1 在 Mull + 夜雨 GhurG	223
		11.1.2 14 VIIII   以小 GIIUI G	223
15	Deb	ian 的技術支援	227
	15.1	參考資料	227
	15.2	查詞意	231

	15.3	查找流行的 Debian 套件	231
	15.4	Debian 錯誤追蹤系統	232
	15.5	郵件列表	232
	15.6	Internet Relay Chat (IRC)	232
	15.7	搜尋引擎	233
	15.8	網站	233
	111 M		
Α	附琢		235
	A.1	作者	235
	A.2	保證	238
	A.3	回饋	238
	A.4	文件格式	238
	A.5	Debian 迷宮	238
	A.6	Debian 谚語	239

# Chapter 1

# 序言

Debian 參考手冊 (http://qref.sourceforge.net/) 的目標在為 Debian 的使用者提供全面性的指導,就如同一本安裝之後的使用手冊。本文假設讀者願意閱讀 shell scripts,並且在閱讀本文件前已具有 Unix-like 系統操作的基礎技能。

我並不打算在文中解釋所有的細節,因爲你可以在 manual 頁面、 info 頁面或 HOWTO 文件中獲得詳細的說明。我希望爲讀者提供更直接實用的資訊,但非全面性的說明。這些資訊包含了在文中列出實際的指令順序,或是在 http://www.debian.org/doc/manuals/ debian-reference/examples/下的 script 範例。在執行命令之前必須先了解這些範例的 内容。這些指令的順序可能需要做些調整以符合你的系統。

文中大部分的技巧提示均引述自權威參考文獻,在'參考資料'227 中列出這些參考書目。

該文件的主旨為 "quick reference" 但内容已經擴充許多。 儘管如此, Keep it short and simple (KISS) 是本文寫作的主導原則。

如果你在尋找系統緊急救援方面的幫助,請立刻閱讀 'Debian 生存指令' 76。

# 1.1 官方文件

本文件最新官方文件是放在 Debian archives 中名為 debian-reference-zh-tw 的套件裡, 你也可在 http://www.debian.org/doc/manuals/debian-reference/取得。

本文件最新開發版本位於 http://qref.sourceforge.net/Debian/ 。該計畫是位於 http://qref.sourceforge.net/ ,在那兒可以下載本文件的純文字格式, HTML 格式、 PDF 格式、 SGML 格式和 PostScript 格式。

## 1.2 文件慣例

"Debian 參考手冊" 透過簡短的 bash 命令提供許多資訊。以下是其文件慣例:

# root 身分所下的命令

\$ 一般帳號身分所下命令

... 命令動作描述

這些 shell 指令的範例使用 PS2=""。和 bash 相關資訊請參閱 'Bash - GNU 標準互動式 shell' 203。

參考:

- bash(1) 表示 Unix manual 頁面資訊。
- info libc 表示 GNU TEXINFO 頁面資訊。
- The C Programming Language 表示書名。
- http://www.debian.org/doc/manuals/debian-reference/表示 URL。
- /usr/share/doc/Debian/reference/表示系統中的檔案。

本文中使用下列的缩寫:

• LDP: Linux Documentation Project (http://www.tldp.org/)

• **DDP**: Debian Documentation Project (http://www.debian.org/doc/)

其他的縮寫在使用前就已經定義在文件內了。

本文中只提供了 LDP 文件的 URL。然而 Debian; 已經包裝了 LDP 的套件,安裝好套件後,可 以在 /usr/share/doc/HOWTO/ 取得這些檔案。

參閱'參考資料'227。

# 1.3 scripts 範例

伴隨 著本文(debian-reference-zh-tw)中的 Example scripts (http://www.debian. org/doc/manuals/debian-reference/examples/) 的範例可以在/usr/share/doc /Debian/reference/examples/套件中取得。隱藏檔開頭的"."被轉換成底線"\_"。如果 有幾個可選擇的範例的話,會加上附加的延伸檔名

### 1.4 基本設定

如果你的系統是最小安裝且希望能善用本文的話,建議執行下面的命令以安裝一些有幫助的文件。

# aptitude install mc less ssh vim kernel-package \
 manpages-dev doc-debian doc-linux-text \
 debian-policy developers-reference maint-guide \
 apt-howto harden-doc debian-reference \
 libpam-doc glibc-doc samba-doc exim4-doc-html

# 1.5 Debian distributions 的基本概念

Debian 同時維護三個不同的發行版本:這些版本是:

- stable: \_\_適用於架設產品化伺服器,該版本只會更新安全性修正的套件。參閱 'stable發行版本'6。
- testing: 工作站的首選,該版本包含了較新版本的桌面軟體以及測試。 參閱 'The testing distribution' 6 。
- unstable: Cutting edge。Debian 開發者的選擇。 參閱 'The unstable distribution' 7。

當unstable的軟體在第一個禮拜或更久沒有 release-critical(RC)的臭蟲時,便會自動移動 到testing。

請參閱 'Debian distribution 開發代號 (codenames)' 7 來了解 Debian 發行版本的代號。 在Sarge發行(2005六月)之前,三個版本的的代號為Woody(stable), Sarge(testing), Sid(unstable)。當 Sarge 發行後,三個版本代號為 Sarge, Etch 和 Sid。當 Etch 發行後,stable和unstable版本將變成Etch和Sid;一個新的testing版本(從stable版本複製)將會產生並賦予一個新代號。

訂閱debian-devel-announce@lists.debian.org這個 mailing list 來取得 Debian 的重 大公告。 請參閱 'Debian archives' 5。

如果您想要使用的軟體版本比目前發行版本提供的還新,那麼您可以跟著'將 distribution 升級 至 stable, testing 或 unstable' 67這一節來升級系統的版本,或者您可只升級該軟體。如 果升級該軟體不是那麼方便或造成更多問題,您可以考慮'把套件引入 stable 系統'83 這一節 所提的 backport 套件。

testing版本會有延誤安裝安全性修正軟體的副作用。這樣的軟體是因為上傳到unstable並 移植到testing的這段期間所造成的延誤。

如果您混用不同的發行版本,例如 testing 和 stable 或 unstable 和 stable,您終究會 安裝到testing或unstable版本的核心軟體,例如libc6,而這樣的行爲無法保證系統無臭 蟲存在。您必須特別小心。

使用testing或unstable的版本會提高遇到嚴重錯誤或臭蟲的機會。這些危險可以利用配置 雙系統(另一個系統為較穩定的 Debian)或者採用'chroot' 115這一節的chroot技巧來避免。 接下來的章節會介紹如何在不同的控制台同時執行不同的 Debian 版本。

在 'Debian 基礎' 5 中我們將講解有關 Debian distribution 的一些基本概念之後,您會學到一 些基礎知識來幫助您與最新的軟體快樂相處,並從 testing 和 unstable distributions 中獲 益。心急的讀者可以直接翻到 'Debian 生存指令' 76 。祝你升級愉快!

# Chapter 2

# Debian 基礎

這一章節是針對非開發者說明 Debian 系統的原理。想要了解更細微詳盡的訊息,請參閱:

- Debian Policy Manual
- Debian Developer's Reference
- Debian New Maintainers' Guide
- 以上文件均列在'參考資料'227。

如果您在尋找比較淺顯易懂的 "how-to",請直接跳到 'Debian 套件管理系統' 71 或其他相關的 章節。

本節的寫作是基於"Debian FAQ"改寫,適合當作一般 Debian 系統管理者的起步文件。

# 2.1 Debian archives

#### 2.1.1 目錄結構

每個被包裝好的 Debian 套件可以從其中一個 Debian mirror site (http://www.debian. org/mirror/)映射站台透過 FTP 或 HTTP 取得。

以下的目錄皆可在 Debian 映射站台中的 debian 目錄找到:

**dists/**: 該目錄存放 "distributions",主要是用來取得不同發行版本的套件。有些舊套件 和Contents-\*.gz 和 Packages.gz 仍然存放在這裡。

**pool/**: 所有 Debian releases 及 pre-releases 的套件的新的實體位址。

tools/: 建立開機片,分割硬碟,解/壓縮檔案,啓動 Linux 的 DOS 工具程式

doc/: 問與答,臭蟲回報等基本 Debian 文件。

**indices/**: Maintainers file 和 override 檔案。

project/: 大多為開發者的資源,例如:

**project/experimental/:** 該目錄存放的套件都是開發中且為 alpha 測試階段。使用者不應抓取這裡的套件,因爲這些套件是危險且會對系統造成傷害。

project/orphaned/: 維護者不再維護且從發行版本移除的"孤兒"軟體放在該目錄。

#### 2.1.2 Debian distributions

stable distribution 套件的入口, Debian Sarge (3.1r0), 被登記到 stable (符號鏈接指向 sarge/目錄):

#### 2.1.3 stable 發行版本

stable版本(Debian Sarge (3.1r0))的stable(連結到sarge/)目錄下紀錄了不同的套件總件:

- stable/main/:該目錄包含了最近發行的 Debian 系統的套件版本。
   這些套件均遵循 Debian Free Software Guidelines (DFSG) (http://www.debian.org/social\_contract#guidelines)(它位於/usr/share/doc/debian/social-contract.txt需安裝 debian-doc),它們均可以自由使用和散布。
- stable/non-free/:經過DFSG的驗證而無法稱為free的套件皆放在該目錄下。
   例如,有些套件的許可證條款 (licenses) 禁止其用於商業的 distribution。有些雖可以再 散布,但本身是共享套件而非自由套件。
- stable/contrib/:這部份的軟體本身是 DFSG-free 但由於某些原因使得必須依賴非 DFSG-free 的軟體才能安裝使用。

除了上述的目錄外,目前的實體套件存在的路徑為 pool 目錄('pool 目錄' 8)。

現階段 stable 版本的臭蟲報告均列在 Stable Problems (http://ftp-master.debian. org/testing/stable\_probs.html) 網頁上。

#### 2.1.4 The testing distribution

testing distribution 的套件入口, Debian Etch, 在 unstable 中通過某種程度的測試後會登記到 testing (符號鏈接指向 etch/) 目錄。目前的實體套件存在的路徑為 pool 目錄 ('pool 目錄'8)。在 testing/下同樣有 main, contrib, 和 non-free 子目錄, 它們的作用與 stable/中的一樣。

這些套件必須可以同時運行於所有架構並且能正常安裝。比起在 unstable 中的對應版本,它們必需有更少的 release-critical 錯誤。這個種方式,我們將 testing 視爲更接近發行的候選版本。有關 testing 機制的更多資訊請參閱 http://www.debian.org/devel/testing。

testing distribution 的最新消息發佈在下列站台:

• update excuses (http://ftp-master.debian.org/testing/update\_excuses. html)

- testing problems (http://ftp-master.debian.org/testing/testing\_probs. html)
- release-critical bugs (http://bugs.debian.org/release-critical/)
- base system bugs (http://bugs.qa.debian.org/cgi-bin/base.cgi)
- bugs in standard and task packages (http://bugs.qa.debian.org/cgi-bin/ standard.cgi)
- other bugs and bug-squashing party notes (http://bugs.qa.debian.org/)

#### 2.1.5 The unstable distribution

unstable distribution 的套件入口,總是被命名為"Sid",被登記到 unstable (符號鏈接指向 sid/) 目錄,上傳至 Debian archive 的套件在被移至 testing/前就一直放在這兒。目前的實體套件存在的路徑為 pool 目錄 ('pool 目錄'8)。在 unstable/下同樣有 main, contrib和 non-free 子目錄,它們的作用與 stable/中相同。

unstable distribution 反映了系統最新的開發進展。歡迎廣大用户使用並測試這些套件,同時也提醒你們這些套件還不完善。使用 unstable distribution 的好處就是你可以獲得 Debian 軟體專案 — 的最新更新,不過新東西也會出新問題,你得好壞兼收:-)

unstable distribution 的最新臭蟲報告見於 Unstable Problems (http://ftp-master. debian.org/testing/unstable\_probs.html) 網頁上。

#### 2.1.6 The frozen distribution

當 testing distribution 足夠成熟了,它便成為 frozen,表示這個版本不再加入新程式,只進 行除錯工作。如果需要的話, dists 目錄中會建立新的 testing 目錄樹,並給予新的開發代 號。 frozen distribution 再經過幾個月的測試、更新、再凍結也稱之為"循環測試"

我們將 frozen distribution 中可能延遲套件或整個版本發佈的錯誤都記錄在案,一但錯誤總數 低於可接受的最大值, frozen distribution 就晉升成 stable,而新版本發布了,前一個 stable distribution 成爲過期版 (obsolete) (並被移至相對應的目錄)。

## 2.1.7 Debian distribution 開發代號 (codenames)

存在於 dists 目錄下的實體目錄名稱,例如 sarge/和 etch/,就是"開發代號 (codenames)"。當某個 Debian distribution 處於開發階段,它並沒有版本號碼,取而代之的是開 發代號。使用開發代號的目的在於簡化建立 Debian distributions 映射站台的工作(例如 unstable的實體目錄突然改名為 stable/,許多文件都沒必要再次下載)。

現在, stable 是一個指向 Sarge 的符號鏈接, testing 是指向 Etch 的符號鏈接。也就是 說 Sarge 是當前的 stable distribution, Etch 是當前的 testing distribution。

unstable/distribution 是指向 sid/的永久符號鏈接,即 unstable distribution 總是稱為 Sid 。

#### 2.1.8 已用過的開發代號

已使用過的開發代號有: "Buzz"用在 release 1.1, "Rex"用在 release 1.2, "Bo"用在 releases 1.3.x, "Hamm"用在 release 2.0, "Slink"用在 release 2.1, "Potato"用在 release 2.2, "Woody"用在 release 3.0 和 "Sarge"用在 release 3.1。

#### 2.1.9 開發代號的來源

到目前為止它們均出自 Pixar 的電影 玩具總動員 (Toy Story)。

- Buzz (Buzz Lightyear) 是個太空人,
- Rex 是隻暴龍,
- Bo (Bo Peep) 是個放羊的女孩,
- Hamm 是個小豬撲滿,
- Slink (Slinky Dog) 是隻玩具狗,
- Potato 當然就是 Potato Head 先生,
- Woody 是個牛仔,
- Sarge 是綠色塑膠玩具士兵的首領,
- Etch (Etch-a-Sketch) 是玩具黑板,
- Sid 是隔壁的男孩,那個玩具終結者。

#### 2.1.10 pool 目錄

過去,套件均放在 dists 目錄下相應的 distribution 子目錄中。這種做法產生了許多問題,例 如映射站台進行新版本發佈時大量頻寬被消耗。

現在套件均依原本的套件名稱分類放進一個巨大的池子"pool"。為了方便管理, pool 目錄下 按屬性再分類 (main, contrib, 和 non-free),分類下面再按原本的套件名稱的英文字首 字母歸檔。這些目錄包含的文件有:運行於各種系統架構的二進位套件,和生成這些二進制套 件的源碼套件。

你可以執行命令 apt-cache showsrc *mypackagename*,查看 "Directory:"行獲知每個套件的存放位置。例如:apache 套件存放在 pool/main/a/apache/。因爲 lib\* 的套件數量 龐大,它們以特殊的方式歸檔:例如,libpaper 套件存放在 pool/main/libp/libpaper /。

諸如 apt 等命令訪問的索引文件仍位於 dists 目錄中。

通常,你大可不必注意這些事情,新版的 apt 和舊版 dpkg-ftp 會自動處理它們。想了 解更多資訊,參閱 RFC: implementation of package pools (http://lists.debian.org/ debian-devel-announce/2000/debian-devel-announce-200010/msg00007. html)。

#### 2.1.11 sid 的歷史記錄

過去 Sid 並不存在, Debian archive 組織有一個主要的工作流程:它假設當前 unstable 中建 立了某個軟體開發項目,它會在整個 distribution 形成一個新的 stable/ 時發佈。但是因爲 很多軟體不是用這個方式在開發,所以一但軟體要發佈時,就必需把整個目錄都搬移到 stable 下。因為在搬移目錄時會用掉大量的頻寬,所以這個流程就顯得很不切實際。

經過幾年的研究摸索, archive 管理員提出一個方案,將未發佈的二進位檔放在一個名為 sid 的特定目錄。當它們首次發布時,將會建立一個從目前 stable 指向 sid 的鏈接,從現在起, 他們會建立在unstable/的目錄下。這個方案的確會使用者覺得困惑。

有了套件 pool 的幫助(參閱 'pool 目錄'8),在 woody distribution 開發過程中,二進位套件均按一定規範存放於 pool 目錄,而與 distribution 無關,當發佈新版本時,就不會再出現大量頻寬被消耗的問題。(不過,開發過程還是會用掉大量頻寬)。

#### 2.1.12 在 incoming 中的上傳套件

上載的套件在經確認它們是由 Debian 開發者上載的後,會先存放於 http://incoming. debian.org/ (對於那些無維護者上載 (Non-Maintainer Upload (NMU))的套件則放入 DELAYED 子目錄)。會有一天,它們將從 incoming/移入 unstable/。

在緊急情況下,你可能會等不及它們移入 unstable/ 而直接從 incoming. 中下載安裝。

#### 2.1.13 找回舊套件

最新的 Debian distribution 存放在任何一個 Debian mirror site (http://www.debian.org/ mirror/)的 debian 目錄下。舊版本的 Debian 如 Slink 存放在 http://archive.debian. org/或 Debian 映射站台的 debian-archive 目錄下。

舊的 testing 和 unstable 套件存放在 http://snapshot.debian.net/。

#### 2.1.14 Architecture sections

在每個主要目錄樹下 (dists/stable/main 、 dists/stable/contrib 、 dists/stable/non-free 、 dists/unstable/main 等) 按晶片架構又分了子目錄,每個子 目錄中存放著在該晶片架構下編譯的二進位套件。

- binary-all/,存放與架構無關的套件,如 Perl script、純文件等。
- binary-platform/,存放運行於該平台的二進制套件。

請注意,二進位套件不再存放在這些目錄中,而是放在最上層 pool 目錄中。但爲了向下相容,所以在目錄中仍保留有索引文件 (Packages 和 Packages.gz)。

要獲得實際上的二進制架構技術支持,請參閱各發行版的發佈記錄(Release Notes)。它們放發佈記錄站台 stable (http://www.debian.org/releases/stable/releasenotes)和 testing (http://www.debian.org/releases/testing/releasenotes)。

#### 2.1.15 源碼

Debian 系統中包含了所有東西的源碼,不僅如此,許可證條款規定系統中所有的程式必須和 其源碼一起發行,或提供源碼。

通常源碼發佈在 source 目錄,該目錄同時處於所有特定平台的目錄中,或是把更新的源碼放 在 pool 目錄中(參閱 'pool 目錄'8)。對於不太熟悉 Debian archive 目錄結構的用户,想獲 得源碼可以試試 apt-get source mypackagename 命令。

有些套件,如著名的 pine,由於許可證限制,只提供源碼套件。(最近,pine-tracker 套件提供了一個簡易的安裝方式。)參閱,把套件引入 stable 系統 83 和,包裝套件 221 教你如何手動編建一個套件。

contrib 和 non-free 目錄中的套件可能不提供源碼,因為它們不算正式 Debian 系統的一部分。

### 2.2 Debian 套件管理系統

#### 2.2.1 Debian 套件概觀

套件通常包含了實現一系列相關命令或特性的所有必需的檔案。Debian 套件有兩種類型:

- Binary packages (二進位套件),它包含可執行檔、設定檔、 man/info 頁面、版權聲明和其它文件。這些套件以 Debian 特定的格式散佈 (參閱 'Debian 套件格式'11);它們通常使用.deb 的延伸檔名來散佈。這種二進位套件可使用 Debian 工具 dpkg 解開,詳情見相關的 manual 頁面。
- Source packages (源碼套件),其中包含一個.dsc 檔用於描述源碼套件(包括下列檔案的名稱),一個.orig.tar.gz 格式的檔案包含了原本未經修改的源碼,以及一個.diff.gz 文件包含了該套件為了符合 Debian 格式對源碼所做的修改。dpkg-source 工具可用於包裝和解開 Debian 源碼套件,詳情可參閱相關 manual 頁面。

使用套件管理系統來安裝套件時會用到由套件維護者建立的"相依資訊"。這些相依資訊記錄在 每個套件的 control 文件中。例如,包含 GNU C 編譯器 (gcc) 的套件"相依"於包含連結器 (linker) 和組譯器 (assembler) 的 binutils套件。如果用户試圖在沒有安裝 binutils 的情況 下安裝 gcc,套件管理系統 (dpkg) 就會顯示錯誤信息,告訴你需要安裝 binutils,並停止 安裝 gcc。(不過,倔強的用户可以對這個信息視而不見,參閱 dpkg(8))。)更多資訊請參 閱下面的'套件相依性'13。

Debian 套件管理工具可用來:

- 操作和管理套件或套件的局部内容,
- 分割套件以助於那些使用有限容量的儲存工具如軟碟傳輸的用户,
- 幫助開發者打包套件,
- 幫助用户從遠端 Debian archive 站台安裝套件。

#### 2.2.2 Debian 套件格式

Debian 的"套件",或是 Debian 保存壓縮檔 (archive) 包含了與特定程式或一組相關的程式有關的可執行檔,函式庫,和文件。通常, Debian 保存壓縮檔是以.deb 做為結尾。1

Debian 二進制套件的格式說明請參閱 deb(5) 的說明文件。由於 deb 的内容是可變的(不同於 各個主要 Debian 版本),所以請使用 dpkg-deb(1) 來控管.deb 檔案。

至少從 Sarge 開始,所有的 Debian 套件檔皆可以透過傳統的 Unix 指令 ar 和 tar 來控制,即 使是 dpkg 不存在也沒關係。

#### 2.2.3 Debian 套件檔名的命名

Debian 套件檔案名稱是採用以下的規則來命名的:

foo\_ver-rev\_arch.deb

通常, foo 是套件名稱, ver 是原作釋出的版本 (the upstream version number), rev 是 Debian 釋出的版本,以及 arch 是目的硬體架構。 檔案是很容易更改,這是當然的喔。 您可以執行以 下指令來查詢該套件檔案名稱的資訊:

dpkg --info filename

Debian 釋出版本是由 Debian 開發者決定或製作該套件的人。 釋出版本的改變就表示套件有些 許的變化。

#### 2.2.4 保留本機設定檔

能被系統管理者修改的檔案都放在 /etc/ 目錄下。 再升級套件時, Debian 政策控制了所有的本機設定檔更新動作。

如果預設的設定檔是套件本身所有,則該檔案稱為"conffile"。從套件安裝後,如果 conffiles 曾被系統管理者修改過,則套件管理系統將不會更新這些 conffiles,除非系統管理者願意。換 句話說,如果 conffile 都未曾修改,則在升級套件的同時也會一併更新 conffile。對大部份的 情況來說,這是值得期許的,因爲方便降低 conffiles 的變更程度。

要列出套件的 conffiles, 請執行下述的指令:

dpkg --status package

<sup>&</sup>lt;sup>1</sup>debian-installer 專案介紹了一種副檔名為.udeb 的套件格式。簡單來說,它是一種簡化型.deb,因為 不用遵循 Debian policy。沒有納入文件且只使用在 debian-installer, Debian installer一開始是針對 Sarge 發 行而設計的。.deb 與.udeb 檔案格式是相同的。udpkg 是用來處理.udeb 套件,但功能上比 dpkg 要來的少 且支援較少的相依資訊。兩者的差別在於 Debian 檔案維護者並不喜歡這些檔案庫裡的.debs 不遵循套件政策, 所以才選擇不一樣的副檔名以免使用者不經意地安裝這些套件到系統上。.udebs 是用在安裝系統時建立的限制 型 Debian 的初始 ramdisk。.udebs are used in an initial ramdisk during the base install only to create a very restricted Debian system.

檔案清單則列在 "Conffiles:" 這一行。

您可以閱讀 Debian Policy Manual 中的 "Configuration files" 部份來取得更多資訊。(參閱 '參考資料' 227)

#### 2.2.5 Debian maintenance scripts

Debian maintenance scripts 是一些在安裝前後自動執行的 scripts。連同 control 檔案,這些 檔案是一個 Debian 檔案中的 "control" 段落的部份之一。

各個檔案説明:

- preinst 該 script 在 Debian 壓縮檔 (.deb) 被解開前才被執行。許多的 "preinst" scripts 將會 停止服務直到套件升級或安裝完成 (完成之後,接下來就是 "postinst" script)。
- postinst 該 script 用在 Debian 壓縮檔 (.deb) 被解開後完成設定套件的手續。"postinst"通常 用來等待使用者輸入或提醒使用者,假如他接受預設的設定值,記得重新設定套件。許 多的 "postinst" scripts 負責在套件安裝或升級後重新啓動服務。
- prerm 該 script 負責停止與套件關聯的daemon服務。它在刪除套件關聯的文件之前執行。
- postrm 該 script 負責修改套件連結或文件關聯或刪除由它建立的檔案。(參閱'虛擬套件'13)。

目前所有的 control 檔案可以在 /var/lib/dpkg/info 目錄下找到。這些檔案關聯於套件 foo 開頭名稱為 "foo" 以及擁有 "preinst", "postinst"...等延伸檔案。該目錄下的 foo.list 紀錄了套件 foo 安裝到系統上的所有檔案。(要注意的是這些檔案的位置是 dpkg 內部運作之 一且易被改變的。)

#### 2.2.6 套件的優先等級

每個Debian套件均被維護者指定了一個優先等級,作為套件管理系統的一個輔助參數,優先等級有:

- Required該級別套件是保証系統正常運作所必須的。
   包含所有必要的系統修護工具。不要刪除這些套件,否則整個系統將受到損壞,甚至無法使用dpkg恢復。只安裝 Required 級別套件的系統不一定能滿足所有的用途,但它可以正常開機,讓系統管理者安裝想要的軟體。
- Important 在任何 Unix-like 系統上均安裝有該級別套件。 系統若缺少這類套件,將會運行困難或不好操作。該級別套件並不包括Emacs或X11或TeX或其它大型應用程式,它們只是一些實現系統底層功能的程式。
- Standard 該級別套件是任何Linux系統的標準,它們組成一個小而精的文字模式的系統。

系統預設安裝就包括了它們。"Standard"級別套件不包括許多大型應用程式,但它包括 Emacs(它比其它應用程式更低階)和Tex及LaTeX的精巧版(不支援X)。  Optional 該級別套件包括那些你可能想安裝的套件,即使對它們並不熟悉或沒有特定的 需求。

它們包括X11,TeX完整發布版和許多應用程式。

 Extra 該級別套件可能與其它較高級別套件衝突,僅當你知道其用途時才會使用它們,或 者它們有特定的需求,這些都使它們不適合定為"Optional"級別。

請注意套件描述中的"Priority: required", "Section: base" and "Essential: yes" 這幾個的差 異性。"Section: base" 是指該套件在建立新系統時一定要安裝的。大部份這類型的套件皆有 "required" 級別(Priority: required)或至少"important" 級別(Priority: import)。且多數被標示 成"Essential: yes"。"Essential: yes" 是指套件管理系統在處理該套件時,必需加入額外的"暴 力"參數,例如使用 dpkg 來移除這類型的套件。舉個例子,libc6, mawk 和 makedev 皆爲 "Priority: required" 和 "Section: base" 但並非 "Essential: yes"。

#### 2.2.7 虛擬套件

虛擬套件是一個統稱,指的一組具有相近功能的套件。舉個例子,tin和trn都是新聞閱讀軟 體,當系統中某個程式需要使用新聞閱讀器工作時就會產會一個關聯關係,它們必須滿足這個 關聯,我們也稱之為它們提供了一個叫做 news-reader 的 "虛擬套件"。

又像是, exim, sendmail 和 postfix都提供郵件傳輸代理的功能。也可稱之爲提供 mail-transport-agent 虛擬套件。如果安裝了其中之一,那麼任何與 mail-transport-agent 有關聯的程式會因爲這個虛擬套件的存在而滿足相依性。

Debian 有個類似的機制,如果系統中提供相同的虛擬套件,系統管理員可以指定一個為首選軟 體。相關的指令是 update-alternatives,更詳細的描述參閱 'Alternative 指令' 86。

#### 2.2.8 套件相依性

Debian套件管理系統有一套套件"依賴性"定義(用一個標誌)用來描述套件需要安裝其他的 軟體才能正常工作或作的更好。

- 套件A依賴(depends)套件B:要運行A必須安裝B。在有些情況下,A不僅依賴B,還 要求B的特定版本。版本依賴通常有最低版限制,A更依賴於B的最新版而非某個特定版本。
- 套件A推薦(recommends)套件B:套件維護者認為所有用户都不會喜歡缺少某些功能的A,而這些功能需要B來提供。
- 套件 A建議(suggests)套件B:B中某些檔案與A的功能相關或加強A的功能。
- 套件A與套件B衝突(conflicts):如果系統中安裝了B那麼A無法運行。"Conflicts"常和"replaces"同時出現。
- 套件A替换(replaces)套件B:B安裝的文件被A中的文件移除和覆蓋了。
- 套件A提供(provides)套件B:A中包含了B中的所有文件和功能。

各個保留字使用方法的更詳細的訊息請參閱Packaging Manual和the Policy Manual。

注意,aptitude和dselect可以對recommends和suggests類 套件進行更細部的控制,apt-get只會簡單的下載安裝depends類套件而不管recommends和suggests>類套件。 這兩個程式均正式使用APT作爲後端程式。

# 2.2.9 何為"pre-depends"

"pre-depends"是一種特殊關聯關系。對於一個普通套件,不管系統中是否存在它的相依套件,dpkg都會將它解包。(解包主要指dpkg從套件中取出安裝文件並放到正確的位置)。如果 套件 Pre-Depends 到其他套件,則這些被 Pre-Depends 的套件會要安裝比的套件先被解包和 設定。<sup>2</sup>這個相依性的用途是爲了提供最小風險的升級。

#### 2.2.10 套件狀態

套件有各種狀態: "unknown", "install", "remove", "purge"和 "hold"。這些 "希望"標記描述了用户打算如何操作這些套件(既可以使用dselect的"Select",也可以直 接使用dpkg)。

它們的意思是:

- unknown 用户並沒描述他想對套件進行什麼操作。
- install 用户希望對套件進行安裝或升級。
- remove 用户希望删除套件,但不想删除任何設定檔。
- purge 用户希望完全删除套件,包括設定檔。
- hold 用户希望套件保持現狀,例如,用户希望保持當前的版本,當前的狀態,當前的一切。

#### 2.2.11 阻止套件升级

有兩種方法阻止套件升級,使用dpkg,或者在Woody中使用APT。

使用dpkg,首先導出套件選擇列表:

dpkg --get-selections \\* > selections.txt

接著編輯文件 selections.txt,修改想要保留的軟體所在的行,例如libc6,則將:

libc6

install

改成:

<sup>&</sup>lt;sup>2</sup>Debina提供這種機制是爲了支援系統從a.out格式向ELF格式安全升級,在該過程中套件的解包順序攸關重要。對於某些重大升級這種機制也很有用。

libc6

hold

儲存設定,將它裝入dpkg資料庫:

dpkg --set-selections < selections.txt</pre>

或者,如果你知道要保留的套件名稱,執行:

echo libc6 hold | dpkg --set-selections

這個命令將在每個套件安裝過程中保持該套件不變。

使用dselect也可以達到同樣的效果。進入[S]elect畫面,找到想阻止其升級的套件,按"=" 鍵(或者"H")。更改在你退出[S]elect畫面後立即生效。

Woody中的APT系統有一個新機制來阻止套件升級,在下載升級檔進程中使用Pin-Priority。參閱幫助頁面apt\_preferences(5),或閱讀http://www.debian.org/doc/manuals/apt-howto/或apt-howto套件。

#### 2.2.12 原始碼套件

原始碼套件發布在source目錄中,既可以手動下載可以使用

apt-get source foo

來抓取它們(參閱apt-get(8)幫助頁面)。

#### 2.2.13 编譯原始碼套件

對於foo套件,從原始碼編譯需要foo\_\*.dsc、foo\_\*.tar.gz和foo\_\*.diff.gz檔案(注意,對於由Debian開發的套件,沒有.diff.gz檔案)。

當你得全了這些檔案,如果你這安裝了dpkg-dev套件,執行:

\$ dpkg-source -x foo\_version-revision.dsc

它將套件解壓到一個名為foo-version的目錄。

使用下列命令建立二進制套件:

\$ cd foo-version

\$ su -c "apt-get update ; apt-get install fakeroot"

\$ dpkg-buildpackage -rfakeroot -us -uc

然後:

# su -c "dpkg -i ../foo\_version-revision\_arch.deb"

安裝新編譯的套件。參閱'把套件引入 stable 系統' 83。

#### 2.2.14 新建Debian套件

建立新套件的詳細訊息,參閱New Maintainers' Guide,該文檔在maint-guide套件中,或 瀏覽http://www.debian.org/doc/manuals/maint-guide/。

## 2.3 Debian系統升級

Debian的設計目標之一就是提供一個一致的升級途徑和安全的升級過程,我們致力於舊版 本的平順升級,套件在升級過程中會將重要訊息提醒管理者,並提供一個可能的解決方 案。你也應該閱讀發布手記(Release Notes),該文檔描述了升級的詳細情況,它存在所 有的Debian光碟中,也可以通過網站訪問http://www.debian.org/releases/stable/ releasenotes或http://www.debian.org/releases/testing/releasenotes。

'Debian 套件管理系統' 71提供了升級的實用指南,本節只作基礎細節的介紹。

#### 2.3.1 dpkg

它是操作套件檔案的主要工具;參閱dpkg(8)獲得完整訊息。 dpkg由幾個原始的輔助程式演化而來。

- dpkg-deb:控制.deb檔案。dpkg-deb(1)
- dpkg-ftp:一個舊的套件取得命令。dpkg-ftp(1)
- dpkg-mountable:一個舊的套件取得命令。dpkg-mountable(1)
- dpkg-split:將大套件分割成小的檔案。dpkg-split(1)

dpkg-ftp和dpkg-mountable已被新的APT系統取代。

#### 2.3.2 APT

APT (the Advanced Packaging Tool) 是Debian套件管理系統的進階介面,由幾個名字以 "apt-"開頭的程式組成。apt-get、apt-cache和apt-cdrom是處理套件的命令列工具,它 們也是其它程式的後端程式,如dselect和aptitude。現在,aptitude是目前首選的系統 管理工具。

更多訊息,可安裝apt和aptitude套件後閱讀aptitude(8)、 apt-get(8)、 apt-cache(8)、 apt-cdrom(8)、 apt.conf(5)、 sources.list(5)、 apt\_preferences(5)。

另一個資源是APT HOWTO (http://www.debian.org/doc/manuals/apt-howto/), 如果安裝了apt-howto包,可在/usr/share/doc/Debian/apt-howto/中找到它。

apt-get upgrade和apt-get dist-upgrade只 抓 取"Depends"類 套 件 , 它 忽 略"Recommend"和"Suggests"類套件,不想這樣的話,可使用dselect。

#### 2.3.3 dselect

這個程式是Debian套件管理系統中以選單驅動的用户介面。特別適用於首次安裝和大幅度升級。參閱'dselect' 74。

更多訊息,請參閱dselect Documentation for Beginners (http://www.debian.org/ releases/woody/i386/dselect-beginner)。

#### 2.3.4 在運行中的系統升級

Debian系統的核心(文件系統)支持替換使用中文件。當升級新套件時如果該套件中包含daemon安裝,則會在該 RunLevel 設定好並重新啓動它們。Debian系統不需要切換到 single-user 模式就能升級運行中的系統了

#### 2.3.5 下載和快取.deb檔案

如果你手動下載套件到磁碟(這完全沒有必要,請閱讀上面有關dpkg-ftp或APT的內容), 當你完成套件安裝工作後,可以從系統中刪除.deb檔案。

如果是使用APT,這些文件會快取在/var/cache/apt/archives/目錄中。你可以在完成安裝後刪除它們(apt-get clean)或者將它們拷貝到另一個機器的/var/cache/apt/archives/目錄中以備以後的安裝。

#### 2.3.6 升級記錄

dpkg會對套件的解開、配置、刪除、完全刪除進行記錄,但不能(目前是這樣)記錄在套件操 作的過程中終端機發生的事情。

最簡單的解決方法是在執行dpkg、dselect、apt-get等工具的對話中加入script(1)程式。

#### 2.4 Debian 開機程序

#### 2.4.1 init 程式

如同所有的Unix一樣,Debian啓動要執行init程式。init的設定檔(/etc/inittab)中指定的第一個執行 script 應該是/etc/init.d/rcS。

而接下來的程序就看 sysv-rc 或 file-rc 套件的安排。假設 sysv-rc 已安裝了(file-rc包含自己的 /etc/init.d/rcS script 並使用該檔案取代 rc 目錄下的軟連結來控制那一個 RunLevel 該啓動那一個服務。)

sysv-rc套件中的 /etc/init.d/rcs 會執行 /etc/rcs.d/ 所有的 script, 為了進行初使化的動作,例如檢查和掛載檔案系統,載入模組,啓動網路服務,設定系統時間...等。然後,也

會執行/etc/rc.boot/下的檔案(除了以'.' 開頭的檔案)也確保相容性。後者的目錄通常是留給系統管理員使用的但並不是很贊成使用它。更多訊息參閱 Debian Policy 說明文件中的'初始 化系統的提示' 119和System run levels and init.d scripts (http://www.debian.org/doc/ debian-policy/ch-opersys#s-sysvinit)。

Debian 不使用 BSD 風格的 rc.local 目錄。

#### 2.4.2 Runlevels

完成系統啓動進程後, init執行預設的 RunLevel。該 RunLevel 由/etc/inittab中的id指定。Debian 使用 id=2。

Debian 使用以下的 RunLevel:

- 1(單人模式 single-user mode),
- 2(各種多人模式 multi-user modes)
- 0(關閉系統),
- 6(重啓系統)。

RunLevel 7,8和9也可以使用,但當套件安裝時,對應他們的 rc 目錄並不會產生。

telinit可用來切換 RunLevel。

當切換 RunLevel 時,/etc/rcrunlevel.d/目錄下的 script 都會被執行。這些 script 檔名的 第一個字母決定該 script 執行的方式: script 以 K 開頭的會夾帶 stop 參數執行。script 以 S 開 頭的會夾帶 start 參數執行; script 以字母順序來執行,所以 "stop" 會提早 "start" scripts 執 行,而跟隨 K 或 S 後的兩個數字決定這些 script 執行順序。

實際上,任意目錄/etc/rc*runlevel*.d/中的script都是指向/etc/init.d/的軟連結。這 些 scripts 都接受參數"restart" and "force-reload";後者是爲了在系統開機後重新啓動服務或 強迫他們重讀設定檔。

例如:

# /etc/init.d/exim4 force-reload

#### 2.4.3 自訂 RunLevel

自訂 RunLevel 是系統管理者工作。底下的建議適用大部份的服務。 Customizing runlevels is an advanced system administration task. The following advice holds for most services.

加入service到 RunLevel R中的方法是建立.. /init.d/service的軟連結在/etc/rcR.d /Sxyservice。xy數字必須該套件安裝時被指定的數字。

更改這個軟連結從開頭是S成K就能取消這個服務,而所填的數字是100減去 xy。

改用 RunLevel 編輯器 sysv-rc-conf 或 ksysv會是比較方便的辦法。

移除特定 RunLevel 目錄下的服務的 S 軟連結也是可行的辦法。這樣作並非取消該服務但卻變成一個"懸浮"狀態直到sysv-rcinit 系統有關連到。更改 RunLevel 後,該服務並不會變成"啓動"或"關閉"的狀態,而是看它先前的設定來決定,不論現在是運行中或不是。注意,雖然該
服務現在是"懸浮"狀態,但當它升級後就會啓動,不管在升級是否在運行或沒有。這是眾所 皆知的 Debian 系統的缺點。注意,您也需要保留該服務的 RunLevel 0和6 的 K軟連結。如果 您刪除該服務所有的軟連結,再升級套件後會回復所有的軟連結到預設的狀態。

強烈建議不要更動/etc/rcS.d/中的任何軟連結。

# 2.5 多樣性支援

Debian提供幾種途徑,在不破壞系統的前提下滿足系統管理員各種要求。

- dpkg-divert,參閱'dpkg-divert 指令'86。
- equivs,參閱'equivs套件'86。
- update-alternative,參閱'Alternative 指令'86。
- make-kpkg可以支援多種啓動載入器。參閱make-kpkg(1)和'核心(重新)編譯'89。

/usr/local/目錄下的所有檔案均屬於系統管理者,Debian不會更動它們。/etc下的大部分 (或全部)檔案屬於conffiles,Debian不會在升級後覆蓋它們,除非系統管理員明確要求 覆蓋。

## 2.6 國際化

Debian系統是國際化產品,不論是在主控台下還是在X下,它都提供多種語言的字元顯示和輸入支援。許多檔案、說明文件以及系統訊息都被翻譯成各種語言。在安裝過程中,Debian提示 用户選擇安裝語言(有時是當地語言的變異)。

如果你安裝的系統提供的語言特性支援不能滿足你的需要,或者你要改變當前語言或安裝別的 鍵盤以適應你的語言,參閱'Localization (l10n)' 150。

## 2.7 Debian和系統核心

參閱'Debian下的Linux核心' 89。

#### 2.7.1 编譯非 Debian 核心的原碼

首先你必須了解Debian policy並遵守標頭文件的規定。

Debian C libraries是依據核心標頭文件最新stable發布版建立的。

例如,Debian 1.2發布版使用版本號為5.4.13的頭文件。它基於所有Linux FTP檔案站點發布的Linux核心源碼包,這些源碼包使用了最新的標頭檔。跟隨核心源碼發布的核心標頭檔位於/usr/include/linux/include/。

如果你編譯某個程序所用的核心標頭檔案比libc6-dev提供的標頭檔案還新,在編譯時 你必須在命令行中加上-I/usr/src/linux/include/。這些情況是存在的,例如,對 於automounter daemon (amd) 套件而言,當新核心改變了對NFS的内部操作方式, amd需 要知道這些改變。這時就需要引用新的核心標頭檔案。

## 2.7.2 自定核心建立工具

對於希望(或必須)使用自定核心的用户,推薦下載kernel-package套件。該套件包含了建 立核心套件的 script。有了它,新建Debian核心套件只需在核心源碼的根目錄執行

# make-kpkg kernel\_image

在核心源碼所在目錄的頂層,執行下述命令可獲得相關幫助

# make-kpkg --help

或閱讀make-kpkg(1) 幫助手冊以及'Debian下的Linux核心' 89。

如果所需的kernel-source-version(version代表核心版本號) 套件不存在,用户就得從Linux檔案站點分別下載最新的(或需要的)核心源碼。Debian的initrd啓動 script 需要一個專門的核心修正才能正確呼叫initrd;參閱http://bugs.debian.org/149236。

有關kernel-package套件的用法詳述見於/usr/doc/kernel-package/README.gz。

#### 2.7.3 模組處理措施

Debian的modconf套件提供了一個shell script (/usr/sbin/modconf),它可以用來自定核 心模組設定。該 script 使用選單界面,用户透過它給出系統中可掛載設備驅動的有關細節,它 再將這些細節訊息生成/etc/modules.conf檔案(其中列出了別名aliases和其它參數,用於 連接各種模組),該設定檔案用來掛載/etc/modutils/目錄下和/etc/modules(其中列 出了需要在系統啓動時掛載的模組)目錄的相關模組。«««< system.sgml

新版的Configure.help提供建立自定核心的步驟和幫助,同樣,modconf套件中也有一系 列幫助檔案(位於/usr/share/modconf/目錄下),告訴你如何對模組設定合適的參數。 =======

新版的Configure.help提供建立自定核心的步驟和幫助,同樣,modconf套件中也有一系 列幫助檔案(位於/usr/share/modconf/目錄下),告訴你如何對模組設定合適的參數。 >>>>> 1.19

## 2.7.4 移除舊版核心套件

kernel-image-NNN.prermscript可用來檢查目前運行中的核心版本,以確定是否與你要移除的核心版本相同。因此你可以使用如下命令刪除不想要的核心套件:

# dpkg --purge --force-remove-essential kernel-image-NNN

(當然,要將NNN替換成你的核心版本和修訂版本。)

# Chapter 3

# Debian 系統安裝提示

安裝 Debian 的官方文件位於 http://www.debian.org/releases/stable/ 和 http: //www.debian.org/releases/stable/installmanual。

文件的開發中版本位於 http://www.debian.org/releases/testing/ 和 http:// www.debian.org/releases/testing/installmanual (處於寫作中,有時可能無法 存取)。

雖然此章節是以 Potato 安裝程式開始寫起,但大部分的內容皆已更新到 Woody 安裝程式且兩 者是非常類似的。由於 Sarge 將會使用一套完全新型的安裝程式,所以請將該文件當作 Sarge 安裝程式的另一個參考。也請注意部份重要的套件已經更換檔名和安裝的優先等級。例如 說,Sarge 已經將預設的 MTA exim 替換成 exim4以及 coreutils 也被其他軟體取代。您需 要做些安裝上調整。

# 3.1 一般Linux安裝提示

如果您在尋找精簡的 Debian 安裝光碟,請別忘記檢查 http://www.debian.org/CD/ netinst/。

使用 testing 或 unstable 版本會增加遇到嚴重軟體錯誤的風險。在機器上安裝一個穩定版本的 Debian ,然後使用多重開機方案可有效控制風險。或者使用更安全的 chroot 技巧,參閱 'chroot' 115 。後者可實現同時在不同的主控台 (console) 執行多種版本的 Debian 。

#### 3.1.1 硬體相容性

Linux 相容絕大多數 PC 硬體,而且幾乎可以安裝在任何系統架構上。對我而言它的安裝過程和 Windwos 95/98/Me 一樣簡單。可相容的硬體清單也在不斷增加。

如果你有一部筆記型電腦,可以去 Linux on Laptops (http://www.linux-laptop.net/) 查找對應品牌和型號的安裝指導。

我建議的桌上型硬體通常「比較保守」:

- 用於工作目的時最好選 SCSI 而不選 IDE , IDE / ATAPI HD 適於個人使用。
- IDE/ATAPI CD-ROM (或 CD-RW)。
- 最好選 PCI 而不選 ISA, 特別是網路卡 (NIC)。
- 選用便宜的 NIC。 Tulip for PCI、NE2000 for ISA 都不錯。
- 初學 Linux 安裝時避免使用 PCMCIA (筆記型電腦)。
- 別使用 USB 鍵盤、滑鼠,... 除非你想證明自己的實力。

如果你的機器特別慢,最好把硬碟插到另一個速度快些的機器上安裝。

### 3.1.2 確定 PC 硬體和晶片組

在安裝過程中,可能會要你確定電腦的硬體或晶片組等。有時了解這些資訊並非易事,這裡有 個方法:

1 打開機殼親自看看。

- 2 記下顯示卡晶片、網路卡晶片、串列介面和 IDE 埠週圍的晶片上印的數字。
- 3 記下印在 PCI 和 ISA 插卡背面的名稱。

## 3.1.3 在 Debian 下確定 PC 硬體

在 Linux 系統中使用下列命令,可獲得機器當前使用的硬體及設定的相關資訊。

- \$ pager /proc/pci
- \$ pager /proc/interrupts
- \$ pager /proc/ioports
- \$ pager /proc/bus/usb/devices

在安裝過程中按 Alt-F2 進入控制台,可執行這些命令獲得幫助。

完成第一次安裝之後,接著安裝額外的套件,例如 pciutils, usbutils和lshw,您就能取 得更多系統的資訊。

\$ lspci -v |pager
\$ lsusb -v |pager
# lshw |pager

典型的中斷用途:

- IRQ0: timer output (8254)
- IRQ1: keyboard controller
- IRQ2: cascade to IRQ8-IRQ15 on PC-AT
- IRQ3: secondary serial port (io-port=0x2F8) (/dev/ttyS1)
- IRQ4: primary serial port (io-port=0x3F8) (/dev/ttyS0)
- IRQ5: free [sound card (SB16: io-port=0x220, DMA-low=1, DMA-high=5)]
- IRQ6: floppy disk controller (io-port=0x3F0) (/dev/fd0, /dev/fd1)
- IRQ7: parport (io-port=0x378) (/dev/lp0)

- IRQ8: rtc
- IRQ9: software interrupt (int 0x0A), redirect to IRQ2
- IRQ10: free [network interface card (NE2000: io-port=0x300)]
- IRQ11: free [(SB16-SCSI: io-port=0x340, SB16-IDE: io-port=0x1E8,0x3EE)]
- IRQ12: PS/2 Mouse
- IRQ13: free (was 80287 math coprocessor)
- IRQ14: primary IDE controller (/dev/hda, /dev/hdb)
- IRQ15: secondary IDE controller (/dev/hdc, /dev/hdd)

對於舊型的非 PnP ISA 介面卡可以從 BIOS 設定 IRQ5, IRQ10 和 IRQ11 爲非 PnP 裝置。

對於 USB 設備,它們在 /proc/bus/usb/devices 中以 Cls=nn 的形式列出:

- Cls=00 : Unused
- Cls=01 : Audio (speaker etc.)
- Cls=02 : Communication (MODEM, NIC, ...)
- Cls=03 : HID (Human Interface Device: KB, mouse, joystick)
- Cls=07 : Printer
- Cls=08 : Mass storage (FDD, CD/DVD drive, HDD, Flash, ...)
- Cls=09 : Hub (USB hub)
- Cls=255 : Vendor specific

如果設備類別 (device class) 不是 255 ,則 Linux 可支援該設備。

## 3.1.4 在其它 OS 下確定 PC 硬體

還可從其它作業系統中獲取硬體資訊:

安裝其它商業版 Linux。當前它們在硬體偵測方面做得比 Debian 好。(不過一但 Sarge 中加入 了 debian-installer,情況就會改變。)

安裝 Windows。用滑鼠右鍵點"我的電腦"在選單中選屬性/設備管理,可獲得硬體設定資訊。記下所有的資源資訊如 IRQ、 I/O 埠地址和 DMA。有些舊的 ISA 卡可能要在 DOS 下設定。

#### 3.1.5 Lilo 迷思

"安裝 Lilo 受限於 1024 磁柱(cylinders)。"大錯特錯!

Debian Potato 之後使用的新版1i10支援lba32。只要主機板的BIOS版本支援lba32,1i10就可以衝破 1024 磁柱的限制。

如果你使用的是舊版lilo.conf,請確定在你的lilo.conf檔案開頭的某處加了一行命令指示系統讀取"lba32"。參閱/usr/share/doc/lilo/Manual.txt.gz。

#### 3.1.6 GRUB

GNU HURD 計畫提供的新的系統開機工具 grub 可使用於Debian Woody系統。

# apt-get update
# apt-get install grub-doc
# mc /usr/share/doc/grub-doc/html/
... read contents
# apt-get install grub
# pager /usr/share/doc/grub/README.Debian.gz
... read it :)

要編輯 GRUB 選單,可編輯/boot/grub/menu.lst檔案。參閱'設定 GRUB 啓動參數' 96獲 取有關啓動參數設定資訊,它和lilo的設定方法不太一樣。

#### 3.1.7 選擇開機軟碟

Potato 時期,在進行普通桌上型電腦安裝時我喜歡用 IDEPCI 磁碟套件。到了 Woody,我愛用 bf2.4 啓動磁碟套件。兩者都使用同一版本的 boot-floppies 來建立開機軟碟。

如果你有一張 PCMCIA 網路卡,你就得使用標準啓動磁碟套件(使用的軟碟數最多,但可用的驅動模組最全)並在PCMCIA設定中設定NIC;別在標準網路設定對話框中設定網路卡。

對於特殊系統,需要建立一張自制救援磁碟,可以用別的機器上編譯好的核心壓縮映射站台替換Debian救援磁碟中名為"linux"的核心映射站台。詳情參閱救援磁碟中的readme.txt檔案。救援磁碟使用 MS-DOS 檔案系統,所以你可以在任何系統下編輯它。這將爲那些使用特殊網路卡的人造福。

到了 Sarge,可用 debian-installer 或 pgi 來建立開機軟碟。

#### 3.1.8 安装系统

這方面的官方文件見於 http://www.debian.org/releases/stable/installmanual 或 http://www.debian.org/releases/testing/installmanual (處於寫作中,有時 可能無法存取)

如果你是用boot-floppies安裝 testing 發行版本,你需要在安裝過程中按 Alt-F2 打開一個 控制台,編輯 /etc/apt/sources.list 檔案,將套件下載入口從stable改爲testing。

在安裝/dev/hda的mbr時,我喜歡將lilo裝在諸如/dev/hda3的位置,這樣做可以減小開機 資訊被覆蓋的風險。

下面是我在安装過程中所做的選擇。

- MD5 passwords "yes"
- shadow passwords "yes"
- Install "advanced" (dselect \*\*) and select
  - 取消emacs (如果它被選中了)、nvi、tex、telnet、talk(d);
    - 選上mc、vim、nano-tiny或elvis-tiny任選一個。 參閱'dselect'74 。即使你是個 Emacs 迷也不要在安裝階段使用它, nano 可满足你的需要。也不要安裝其它龐大 的套件如 TeX (Potato曾在此階段設定它為預設安裝)。參閱'應急的編輯器'189了 解此時安裝nano-tiny或elvis-tiny的原因。

• 在每個套件安裝對話框中,回答所有的設定提問均="y"(替換當前值)

• exim: 選第2項,因為我使用ISP的SMTP伺服器發送郵件。

有關dselect的更多資訊,參閱'dselect'74。

#### 3.1.9 網路所需的主機名和IP地址

LAN設定的例子(C subnet: 192.168.1.0/24):

```
Internet
     +--- 外部ISP提供POP服務(使用fetchmail存取)
    Access point ISP提供DHCP服務和SMTP中繼服務
     :
    Cable modem
                     (Dialup)
     LAN網關外部埠: eth0 (IP由ISP的DHCP提供)
   使用老式筆記型電腦(IBM Thinkpad, 486 DX2 50MHz, 20MB RAM)
   執行Linux 2.4核心,提供ext3檔案系統支援
   執行"ipmasg"套件(加入更好的修正,NAT 和 firewall)
   執行"dhcp-client"套件設定eth0 (覆蓋DNS的設定)
   執行"dhcp"套件設定eth1
   執行"exim"作爲smarthost (mode 2)
   執行"fetchmail"設一個較長的時間值(fallback)
   執 行"bind" 作 爲 暫 存 名 稱 伺 服(cache nameserver)器 , 在 從LAN連
入 Internet 時
            作為認證網域名稱伺服器 (authoritative nameserver), 在從LAN中
連入網中某個域時
   執行 "ssh" 使用埠 22 和 8080 (從任何地點連接)
   執行 "squid" 作為代理伺服器來快取 Debian 檔案庫 (APT需要連接它)
  LAN通訊閘内部埠: eth1 (IP = 192.168.1.1, 固定)
          +--- LAN Switch (100base T) ---+
  LAN中一些固定IP客户端
                           LAN中一些DHCP客户端
  (IP = 192.168.1.2-127, 固定) (IP = 192.168.1.128-200, 動態)
```

參閱'網路設定'161了解更多網路設定方面的資訊。參閱'建立通訊閘路由器'184 了解更多區域網路通訊閘設定方面的資訊。

#### 3.1.10 使用者帳號

為了讓機器存取起來有一致的感覺,我的系統中開頭幾個帳號通常是不變的。

我首先建立的使用者帳號名稱是"admin"(uid=100)。我透過該帳號轉發所有的 root email。該帳號加入到adm使用者組(參閱"為何 GNU su 不支援 wheel 群組" 122),這個使用者組為成員提供大量root特權的命令,透過su或sudo命令就能使用PAM。詳情參閱'增加帳號' 42。

#### 3.1.11 建立檔案系統

#### 硬碟分匾

我更喜歡將不同的目錄樹分別裝在不同的分區下,這樣可以將系統崩潰造成的損失減到最小。 例如:

/	==	(/ + /boot + /bin + /sbin)
	==	50MB+
/tmp	==	100MB+
/var	==	100MB+
/home	==	100MB+
/usr	==	700MB+ with X
/usr/local	==	100MB

/usr目錄的大小主要取決於 X Window 應用程式和文件的數目。如果只執行控制台終端/usr /可以小到300MB,但如果你裝了大量Gnome應用程序2GB-3GB也很正常。當/usr/增加得太 大,將/usr/share/移到別的分區是最有效的解救方法。對新的Linux2.4核心包,/目錄的大 小可能要超過200MB。

例如,當前我的 Internet 通訊閘(gateway)伺服器的硬碟使用情況如下(由df -h命令輸出):

archives
d
/

(有塊大空間留給/var/spool/squid作為下載套件時代理伺服器的緩衝區)

下面的fdisk -1輸出提供了一個分區分配方案:

# fdisk -1 /dev/hda # comment
/dev/hda1 1 41 309928+ 6 FAT16 # DOS
/dev/hda2 42 84 325080 83 Linux # (not used)

/dev/hda3	*	85	126	317520	83	Linux # Main
/dev/hda4		127	629	3802680	5	Extended
/dev/hda5		127	143	128488+	82	Linux swap
/dev/hda6		144	157	105808+	83	Linux
/dev/hda7		158	171	105808+	83	Linux
/dev/hda8		172	253	619888+	83	Linux
/dev/hda9		254	335	619888+	83	Linux
/dev/hda10		336	417	619888+	83	Linux
/dev/hda11		418	629	1602688+	83	Linux

裡面有一個未使用的分區。留作安裝第二個Linux或爲增長的目錄樹提供擴充餘地。

# 掛載檔案系統

下面的/etc/fstab文件可完成上述分區的掛載工作:

# /etc/f #	stab: sta	atic filesyste	em informat	cion.			
# filesy	stem m	ount point	type o	options d	lump	pas	S
/dev/hda	.3	/	ext2	defaults, errors=remount	rc	0 1	
/dev/hda	.5 1	none	swap	SW	0	0	
proc		/proc	proc	defaults	0	0	
/dev/fd0		/floppy	auto	defaults,user,noauto	0	0	
/dev/cdr	om .	/cdrom	iso9660	defaults, ro, user, noauto	<i>,</i> 0	0	
#							
# keep p	artition	s separate					
/dev/hda	.7	/home	ext2	defaults	0	2	
/dev/hda	.8	/var	ext2	defaults	0	2	
/dev/hda	.6	/var/lib/cvs	ext2	defaults	0	2	
<pre># noatim</pre>	e will s	peed up file a	access for	read access			
/dev/hda	.9	/usr	ext2	defaults,noatime	0	2	
/dev/hda	.10	/var/cache/apt	/archives	ext2 defaults	0	2	
# very b	ig parti	tion for proxy	v cache				
/dev/hda	.11	/var/spool/squ	uid ext2	rw	0	2	
# backup	bootable	e DOS					
/dev/hda	.1 .	/mnt/dos	vfat	rw,noauto	0	0	
# backup	bootable	e Linux system	n (not done	e)			
/dev/hda	.2	/mnt/linux	ext2	rw,noauto	0	0	
#							
# nfs mo	unts						
mickey:/		/mnt/mickey	nfs	ro,noauto,intr	0	0	

goofy:/ /mnt/goofy nfs ro,noauto,intr 0 0
# minnie:/ /mnt/minnie smbfs ro,soft,intr,credentials={filename} 0 2

對於NFS,我使用noauto、intr以及預設的hard項。如果有死連接,掛載進程可能會死掉,可以使用 Ctrl-C 恢復。

對於使用Samba連接的Windows機器(smbfs), rw, auto, soft, intr是個好方案,參閱'Samba設定'35。

對於軟驅,使用noauto,rw,sync,user,exec可以防止因退盤前未執行卸載命令而造成文件 損壞,但它會降低寫盤速度。

#### 自動掛載

自動掛載的要點:

• 加載vfat模組,從而/etc/auto.misc可使用-fstype=auto參數:

# modprobe vfat # prior to the floppy access attempt

- ... or to automate this settings,
- # echo "vfat" >> /etc/modules
- ... 重啓系統。
- 設定/etc/auto.misc如下: floppy -fstype=auto,sync,nodev,nosuid,gid=100,umask=000 :/dev/fd0
   ... 此處gid=100表示 "users"。
- 在 /home/user中建立連結檔案cdrom和floppy,分別指向/var/autofs/misc/cdrom和/var/autofs/misc/floppy。
- 將user加入 "users" 使用者組。

#### 掛載NFS

外部Linux NFS伺服器(goofy)處於防火牆(gateway)之後。在我的LAN內部,安全策略非 常寬鬆,因爲只有我自己一個使用者。爲了存取NFS,NFS伺服器端要按如下方式加上/etc /exports:

# /etc/exports: the access control list for filesystems which may be
# exported to NFS clients. See exports(5).
/ (rw,no\_root\_squash)

除安裝和啓動NFS伺服器及客户端的套件之外,這一步對啓動NFS伺服器也是必要的。

我通常建立一個單獨的2GB分區,有時想玩玩別的Linux系統,就可以裝在這兒。兩個系統可 共享swap和/tmp分區。多分區方案很難滿足這種需求。如果只想裝一個執行簡單控制台模式 的系統,分區留500MB就綽綽有餘了。

### 3.1.12 DRAM記憶體指導

下列是有關DRAM的簡單指導。

- 4MB: Linux核心執行的最低設定。
- 16MB: 控制台系統執行的最低設定。
- 32MB: 簡單X系統執行的最低設定。
- 64MB: GNOME/KDE系統執行最低設定。
- 128MB: 流暢執行GNOME/KDE。
- 256MB(或更多): 如果您預算充足,何不增加呢?DRAM 很便宜了。

使用啓動選項mem=4m(或lilo append="mem=4m")可以看看只用4MB記憶體時系統如何執行。如果機器的BIOS比較舊且記憶體數大於64MB啓動時就得加lilo啓動參數。

#### 3.1.13 Swap空間

我按下面的指導原則來劃分 swap 空間:

- 每個 swap 分割區均<128MB(使用舊版2.0核心),<2GB(使用最新核心)
- 總容量 = (機器記憶體大小的1到2倍) 或(128MB到2GB之間)
- 將它們分散於不同的硬碟,在/etc/fstab中可使用sw,pri=1選項掛載它們。這樣可確 保核心以RAID方式使用swap分區,最大限度地發揮swap的性能。
- 如果可能,使用硬碟中間的磁區。

盡管你不一定真的需要,但爲系統設定多一些 swap 空間(128MB)總要好點,至少執行有記 憶體漏洞的壞程序,系統會先慢下來而不是馬上死當。

# 3.2 Bash 設定

我按自己的喜好修改shell 啓動 script:

换成你的喜好
保持發布版設定 ( \w -> \W)
换成你的喜好
换成你的喜好
换成你的喜好
换成你的喜好, 會改變所有使用者的設定
换成你的喜好, 會改變所有使用者的設定
换成你的喜好, 會改變所有使用者的設定

可 以 參 考 我 提 供 的 script 範 例 (http://www.debian.org/doc/manuals/ debian-reference/examples/)。我 喜 歡 系 統 看 起 來 清 清 楚 楚 , 所 以 將umask設 爲002或022。

PATH依次由以下設定文件設定:

/etc/login.defs - 在shell之前設定PATH /etc/profile (會呼叫/etc/bash.bashrc) ~/.bash\_profile (會呼叫~/.bashrc)

# 3.3 滑鼠設定

## 3.3.1 PS/2滑鼠

對於ATX主機板上使用PS/2介面的滑鼠,信號通路如下:

mouse -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X

這裡,建立指向/dev/gpmdata的符號鏈接(symlink)/dev/mouse,有助於簡化設定操作。 (例如,如果你決定不再使用 gpm服務,只需在移除gpm 服務,將符號鏈接/dev/mouse指向 /dev/psaux。)

信號通路使得對鍵盤和滑鼠插拔操作,只要重啓qpm就可以生效,不必重啓X!

處於gpm輸出與X輸入之間的信號通路協定有兩種執行方式, "ms3" (使用Microsoft 3button serial mouse協定)和 "raw" (對於已連接的滑鼠使用相同的協定),此處選擇的 協定將決定X設定中使用的協定。

下面我將示範一下如何設定Logitech 3-button(傳統Unix風格的滑鼠)PS/2滑鼠:

如果新版X4不支援你的顯卡則需要使用舊版X3(如某些ATI 64位卡),請設定/etc/X11/XF86Config而不是/etc/X11/XF86Config而不是/etc/X11/XF86Config-4,下面的示例中使用的是X4:

### 使用ms3協定方式

/etc/gpm.conf	/etc/X11/XF8	6Config-4	
device=/dev/psaux	+=====================================	utDevice"	
responsiveness=	Identifier	"Configured	Mouse"
repeat_type=ms3	Driver	"mouse"	
type=autops2	Option	"CorePointe:	r"
append=""	Option	"Device"	"/dev/mouse"
sample_rate=	Option	"Protocol"	"IntelliMouse"
	EndSection		

如果使用這種方式,設定滑鼠類型只需編輯gpm.conf,X的設定將保持不變。參閱我的 script 範例 (http://www.debian.org/doc/manuals/debian-reference/examples/)。

## 使用raw協定方式

```
/etc/gpm.conf
                  | /etc/X11/XF86Config-4
_____
device=/dev/psaux | Section "InputDevice"
                  | Identifier "Configured Mouse"
responsiveness=
                  | Driver "mouse"
repeat_type=raw
                            "CorePointer"
                  | Option
type=autops2
                            "Device" "/dev/mouse"
append=""
                  | Option
                  | Option "Protocol" "MouseManPlusPS/2"
sample_rate=
                   | EndSection
```

如果使用這種方式,設定滑鼠類型可編輯gpm.conf,同時也會改變X的設定。

### 如何設定不同的滑鼠

gpm的設備類型autops2可以自動偵測出市面上大多數 PS/2 滑鼠。不幸的是它也不是萬能的,而且Woody之前的發布版中並不包括它,這時可在gpm.conf中試試ps2或者imps2。想 看看gpm支援哪些類型的滑鼠可輸入:gpm -t help。參閱gpm(8)。

如果使用的是2鍵PS/2 滑鼠,可選上X協定的Emulate3Buttons項。協定中2鍵滑鼠和3鍵 滑鼠的區別在於每次按下中間鍵時,是自動偵測還是自動模擬信號傳給qpm。

對於在X協定'使用raw協定方式' 33或不使用qpm,可使用下面的設定:

- IntelliMouse: 串列埠滑鼠(gpm轉換器使用"ms3")
- PS/2: PS/2 滑鼠(通常首先試試這項)
- IMPS/2: 任何PS/2滑鼠(2鍵、3鍵或滾軸滑鼠)
- MouseManPlusPS/2: Logitech PS/2滑鼠
- . . .

更多資訊可瀏覽Mouse Support in XFree86 (http://www.xfree86.org/current/mouse. html)。

典型的 Microsoft 滾軸滑鼠,有報導說這樣設定最好:

/etc/gpm.conf	/etc/X11/XF8	6Config-4 
device=/dev/psaux	Section "Inp	utDevice"
responsiveness=	Identifier	"Configured Mouse"
repeat_type=raw	Driver	"mouse"
type=autops2	Option	"CorePointer"
append=""	Option	"Device" "/dev/mouse"
sample_rate=	Option	"Protocol" "IMPS/2"
	Option	"Buttons" "5"
	Option "ZA	xisMapping" "4 5"
	EndSection	

對於某些最新的 Toshiba 超薄筆記型電腦,在 System-V init 腳本中設定gpm先於PCMCIA 啓動 可防止系統鎖死。聽起來古怪不過是真的。

## 3.3.2 USB 滑鼠

請確定你設定了所有必需的 Kernel 選項,並在編譯核心時編入核心或模組:

- 在 "Input core support" 選單下:
  - "Input core support" (CONFIG\_INPUT, input.o),
  - "Mouse support" (CONFIG\_INPUT\_MOUSEDEV, mousedev.o),
- 在"USB support" 選單下:
  - "Support for USB" (CONFIG\_USB, usbcore.o),
  - "Preliminary USB device filesystem" (CONFIG\_USB\_DEVICEFS),
  - "UHCI" or "OHCI" (CONFIG\_USB\_UHCI || CONFIG\_USB\_UHCI\_ALT || CON-FIG\_USB\_OHCI, usb-uhci.o || uhci.o || usb-ohci.o),
  - "USB Human Interface Device (full HID) support" (CONFIG\_USB\_HID, hid.o), and
  - "HID input layer support" (CONFIG\_USB\_HIDINPUT)

這邊的小寫字串爲模組名稱。

如果你沒使用devfs,則需按下面的方法建立一個設備節點/dev/input/mice:

```
# cd /dev
# mkdir input
# mknod input/mice c 13 63
```

對於典型的滾軸USB滑鼠,設定如下:

```
| /etc/X11/XF86Config-4
/etc/gpm.conf
_____
device=/dev/input/mice | Section "InputDevice"
responsiveness= | Identifier "Generic Mouse"
                       Drive_
Option
Option "Device
Option "Protocol" "In
"Buttons" "5"
-isMapping" "
                      | Driver
repeat_type=raw
                       | Option
                                    "SendCoreEvents" "true"
type=autops2
append=""
                                    "Device" "/dev/input/mice"
sample_rate=
                                    "Protocol" "IMPS/2"
                        Option "ZAxisMapping" "4 5"
```

更多資訊可瀏覽the Linux USB Project (http://www.linux-usb.org/)。

## 3.3.3 觸控版 (Touchpad)

盡管觸控版在筆記型電腦上預設是模擬2鍵 PS/2 滑鼠行為,仍可從tpconfig套件獲得對這類設備的完全操控。例如在/etc/default/tpconfig中設定OPTIONS="--tapmode=0"可

取消討厭的"click by tap"行為。按如下方法設定/etc/gpm.conf可以控制台下使用觸控版和 USB 外接滑鼠。

```
device=/dev/psaux
responsiveness=
repeat_type=ms3
type=autops2
append="-M -m /dev/input/mice -t autops2"
sample_rate=
```

# 3.4 NFS設定

安裝NFS要設定/etc/exports。

詳情見我的 script 範例 (http://www.debian.org/doc/manuals/debian-reference/ examples/).

# 3.5 Samba設定

參考資料:

- http://www.samba.org/
- samba-doc package

以"share"方式安裝 Samba 比較容易,因爲安裝過程會建立WfW-type共享驅動器。但最好使用"user"模式來設定它。

可以用debconf或vi來設定Samba:

```
# dpkg-reconfigure --priority=low samba # in Woody
# vi /etc/samba/smb.conf
```

詳情參閱\我的 script 範例 (http://www.debian.org/doc/manuals/ debian-reference/examples/).

可透過smbpasswd向 smbpasswd 檔案新增一個新使用者:

\$ su -c "smbpasswd -a username"

確保使用的加密密碼有恰當的相容性。

按下表中相應的值設定os level (數字越大,伺服器的優先級越高):

```
0: Samba with a loose attitude (will never become a master browser)
1: WfW 3.1, Win95, Win98, Win/Me?
16: Win NT WS 3.51
17: Win NT WS 4.0
32: Win NT SVR 3.51
33: Win NT SVR 4.0
255: Samba with mighty power
```

確認使用者是目錄所屬組的成員,並且對目錄賦予了執行權限。

# 3.6 印表機設定

傳統列印方法是lpr/lpd。新的 CUPS 系統 (Common UNIX Pinting System) 提供 了另一方法PDQ,詳情參閱Linux Printing HOWTO (http://www.tldp.org/HOWTO/ Printing-HOWTO.html)。

#### 3.6.1 lpr/lpd

For the lpr/lpd的列印緩衝池(type spoolers)(lpr、lprng和gnulpr),如果它們連接的是PostScript或text-only印表機(最基本的印表機)可按如下方式設定/etc/printcap:

```
lp|alias:\
    :sd=/var/spool/lpd/lp:\
    :mx#0:\
    :sh:\
    :lp=/dev/lp0:
```

上述各行的意思是:

- Head line: *lp*-spool名稱, *alias* = alias
- mx#0 不限最大文件尺寸
- sh 不列印頁眉
- lp=/dev/lp0-本地機印機,或port@host遠程印表機

如果連接的是台PostScript印表機,上述設定就夠用了。如果是Windows機器透過Samba列印,上述設定也適用於任何Windows支援的印表機(不支援雙向通信)。你必須在Windows環境中對印表機做相應的設定。

如果你沒有PostScript印表機,就得用gs安裝過濾系統。有很多自動設定工具可用於設定/etc/printcap,可選擇下列任何一組:

• gnulpr、(lpr-ppd)和printtool— 我就是用這種

- lpr和apsfilter
- lpr种magicfilter
- lprng和lprngtool
- lprng和apsfilter
- lprng和magicfilter

想執行GUI設定工具如printtool,需要 root 權限,參閱'在 X 裡取得 root' 138。printtool可建立列印緩衝池,它使用gs模擬PostScript印表機,所以要使 用PostScript列印驅動來存取它們。在Windows端, "Apple LaserWriter"是標準件。

## 3.6.2 CUPS<sup>TM</sup>

安裝 Common UNIX Printing System (or CUPS™)所有的套件,請開啓 aptitude 並選擇: "Tasks" -> "Servers" -> "Print Server". (Sarge) 為了最好的結果,您必須設定 aptitude 如下: "F10" -> "Options" -> "Dependency handling" -> "[X] Install Recommended packages automatically".

KDE 和 Gnome 桌面系統提供了簡易的印表機設定。如果有安裝 swat,您也能使用瀏覽器來 設定:

\$ mybrowser http://localhost:631

舉個例子,將你的印表機連接到可存取印表機清單中:

- 在主頁面上點擊 "Printers",然後點 "Add Printer",
- 輸入使用者名和密碼,進入"root",
- 按提示新增印表機,
- 返回"Printers"頁面,點選"Configure Printer",
- 設定列印紙尺寸、解析度和其它參數。

更多資訊可瀏覽http://localhost:631/documentation.html和http://www.cups. org/cups-help.html。

# 3.7 桌上型電腦使用CRON

Vixiecron預設會自動安裝,它可用來排定工作。如果你的電腦不是24/7開機的話,並無多大的用處。對桌上型來說,你或許會安裝anacron來取代cron以解決這樣的問題。fcron也是 另一套值得推薦的軟體

參閱'日程安排 (cron, at) '111的説明來設定 CRON 工作。

# 3.8 其它主機安裝提示

#### 3.8.1 初始化安装完成後再装些什麼

到現在,你已經擁有一個小巧但功能頗強的Debian。接下來,可以安裝那些較大的套件了。

在此,你最想做的事就是選擇鐘愛的編輯器和aptitude需要的程序。你可以同時安裝Emacs的多個變種。參閱'流行的編輯器' 189。

然後,再次執行aptitude,選擇(幾乎)所有在"Tasks" -> "End-user"下的套件以取得完整的使用者導向的系統。我通常會編輯/etc/inittab簡化關機步驟。

```
# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -h now
...
```

### 3.8.2 模組

. . .

在初始安裝期間可進行設備驅動模組設定。以後還可使用modconf進行設定,它是基於選單介面的工具,可用來設定那些在初始安裝時未曾設定的模組或完成新核心安裝後的設定工作。

所有預加載(preloading)模組的名稱要加入到/etc/modules清單。也可以使用1smod和depmod進行手工操控。

如果你沒打開IP偽裝(FTP等),別忘了在/etc/modules中加上幾行。

## 3.8.3 CD-RW基本步驟

使用 IDE CD-RW 和 kernel 2.4 的話,請編輯下列文件:

/etc/lilo.conf	(新增append="hdc=ide-scsi ignore=hdc",
	執行 lilo 啓動)
/dev/cdrom	(建立鏈接 # cd /dev; ln -sf scd0 cdrom )
/etc/modules	(add "ide-scsi"和"sg"。如果需要可再加上"sr"。)

詳情參閱'燒錄器' 124。

#### 3.8.4 大記憶體和關機自動斷電

編輯/etc/lilo.conf,設定啓動提示參數如下,實現識別大記憶體(適用於2.2版核心)和 關機自動斷電(適用於APM):

append="mem=128M apm=on apm=power-off noapic"

執 行lilo完 成 上 述 設 定 。 對 稱 多 處 理 器 核 心 (SMP-kernel) 需 要apm=power-off和noapic才能避免我那堆糟糕的 SMP 硬體出問題。系統啓動時, 在啓動提示符後直接輸入這些參數效果也一樣。參閱(其它用於啓動提示的技巧'95。

在 Debian 的2.4版核心中 APM 是作爲模組編譯的,對於這種情況,可在系統啓動後執行# insmod apm power\_off=1或設定/etc/modules:

# echo "apm power\_off=1" >>/etc/modules

還可以這樣:編譯新版核心時加入 ACPI 支援可達到同樣的效果,而且這種方式更適合SMP(只有較新的主機板才支援ACPI)。對於較新的主機板2.4版核心可以直接檢測到大記憶體。

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

在/etc/modules中按如下順序新增參數:

ospm\_busmgr ospm\_system

或者重新編譯核心,在設定時將上述核心選項均設為"y"。總之,有了ACPI支援就不再需要 任何啓動提示參數。

## 3.8.5 無法存取某此站台的怪問題

新的Linux核心預設開啓ECN,對於某些使用劣質路由器的站台,會出現無法存取的問題。檢查ECN狀態可使用:

```
# cat /proc/sys/net/ipv4/tcp_ecn
... 或
# sysctl net.ipv4.tcp_ecn
```

将它关闭:

```
# echo "0" > /proc/sys/net/ipv4/tcp_ecn
... 或
# sysctl -w net.ipv4.tcp_ecn=0
```

每次啓動時禁止TCP ECN,可編輯/etc/sysctl.conf,加上:

 $net.ipv4.tcp_ecn = 0$ 

## 3.8.6 PPP 撥號設定

安裝pppconfig套件,設定 PPP 撥號存取。

# apt-get install pppconfig
# pppconfig
... 依照提示設定 PPP 撥號
# adduser user\_name dip
... 允許user\_name進行 PPP 撥號存取

使用者 (user\_name) 進行 PPP 撥號存取:

\$ pon ISP\_name # 開始 PPP 存取,接通你的ISP
... enjoy the Internet
\$ poff ISP\_name # 停止 PPP 存取, ISP\_name可選

詳情參閱'設定 PPP 裝置'165。

## 3.8.7 /etc/中的其它設定文件

Debian 標準安裝不包含 /etc/cron.deny 文件,想新增可拷貝 /etc/at.deny。

# Chapter 4

# Debian指南

本章爲那些初涉 Debian 世界的新手指出了基本方向。如果你接觸 Unix-like 系統已有一段時間,就能了解我底下說明的部份。請將此文件當成實戰測試吧。

# 4.1 準備開始

從安裝 Debian 到電腦之後,您需要學會一些技巧才能活用它。讓我們給您一些簡單的練習。

## 4.1.1 用 root 登入 shell 提示符號

在開機的過程中,您會看到圖形或文字介面的登入畫面(基於安裝的套件)。為了簡化說明,如 果您看到的是圖形登入介面,請按下 Ctrl-Alt-F1<sup>1</sup> 切換到文字介面的登入模式。 假設您的主機名稱為 foo,那麼登入提示符號則類似:

foo login:

輸入 root,按下 Enter 後並輸入安裝過程設定的密碼。Debian 系統跟 Unix 一樣,密碼是分辨大小寫的。接下來系統會出現一個問候訊息並留下 root 提示符號等您輸入指令。<sup>2</sup>

foo login: root
Password:
Last login: Sun Oct 26 19:04:09 2003 on tty3
Linux foo 2.4.22-1-686 #6 Sat Oct 4 14:09:08 EST 2003 i686 GNU/Linux

Most of the programs included with the Debian GNU/Linux system are freely redistributable; the exact distribution terms for each program

<sup>&</sup>lt;sup>1</sup>同時按下鍵盤左邊的 Ctrl, Alt 以及 F1。

<sup>&</sup>lt;sup>2</sup>如果您編輯 /etc/motd,則問候訊息就會不同。

are described in the individual files in /usr/share/doc/\*/copyright

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

root@foo:root#

透過 root 命令提示符號,您就準備進行系統管理了。root 帳號稱爲超級使用者(superuser)或特 權使用者(privileged user)。使用 root 帳號,您可以達成:

- 讀、寫和移除任何系統上的檔案而不需考慮權限
- 能夠設定系統上任何檔案的擁有者和權限
- 設定任何系統非特權使用者的密碼
- 登入到任何帳號而無需輸入密碼

分享 root 密碼來取得 root 帳號是非常不好的方法。請使用如 sudo(8) 的軟體就是一個較好的 方式來分享管理者權限。

先登入非特權帳號再進行管理動作是值得學習的 Unix 習慣。等需要 root 權限時再使用 sudo, super或su -c來取得特定的功能。請參閱'更安全的工作環境-sudo' 122。<sup>3</sup>

## 4.1.2 針對新手設定的最精簡環境

我認爲學習電腦系統就像是學習一套外國語言。雖然學習指南很有幫助,但您需要去練習並使用這些工具。這個情況下,我建議額外安裝一些套件來輔助,例如 mc, vim, lynx, doc-linux-text 和 debian-policy。<sup>4</sup>

# apt-get update
...
# apt-get install mc vim lynx doc-linux-text debian-policy
...

如果您已經安裝這些套件,則執行上述指令就不會重新安裝。

## 4.1.3 增加帳號

系統安裝完畢,通常會添加一個普通用户帳號,而該帳號會收到寄給 root 的信件。<sup>5</sup>如果您不 想使用這個特殊帳號來練習以下的指令則必須建立一個新的使用者帳號。

如果新的帳號為 penquin, 請輸入:

root@foo:root# adduser penguin

... 回答完所有的問題

<sup>&</sup>lt;sup>3</sup>我必須説明我常常使用超級使用者帳號,只因為它是很容易且我比較不拘小節。

<sup>&</sup>lt;sup>4</sup>安裝 qpm, emacs21 和 doc-linux-html 也是不錯的選擇喔,請參閱 '滑鼠設定' 32 and '編輯器' 189。

<sup>&</sup>lt;sup>5</sup>我傾向將此帳號命名爲 admin,當然也可以是任何有權威性的名字。

就會建立該帳號了。6在進行下面的步驟前,我們先學習一些基礎。

#### 4.1.4 切換虛擬主控台

在未調整過的 Debian 系統中,總共有六個獨立的虛擬終端機。也就是說您可以在文字介面的 主控台中使用六個可切換的 VT-100 終端機。按下左邊的 Alt 加上 F1-F6 可以切換到另外一個 終端機。每一個虛擬終端機允許不同的帳號登入且不受其他終端機影響。多人環境是 Unix 偉 大的設計且容易讓人上癮。

如果您不小心在執行 X 視窗系統的機器按下 Alt-F7,則主控台螢幕會切換到視窗介面,按下 Ctrl-Alt-F1 就能回到文字主控台。試著練習切換不同的主控台並跳回到原來這一個來習慣這些 特性。

#### 4.1.5 如何關機

和其他現代的作業系統一樣,系統檔案會快取到記憶體中,在切斷電源之前 Debian 系統需要 完成特定的關機程序才能維持檔案的完整性。使用 root 提示命令列來執行以下的關機指令:

# shutdown -h now

這是多用户模式下的關機命令。如果是單一使用者模式,請執行以下的命令:

# poweroff -i -f

您也能按下 Ctrl-Alt-Delete 來關機。7

等待螢幕上顯示 "System halted" 之後切斷電源。如果在BIOS和Linux中開啓了 APM 或 ACPI,系統會自動切斷電源。詳情參閱'大記憶體和關機自動斷電' 38。

#### 4.1.6 把玩時間

現在您可以大膽地使用 Debian 系統而不用擔心任何危險,因爲您使用的是非特權帳號: penquin。<sup>8</sup>

我們來試著登入 penguin,如果您在 root 命令提示列,請按下 Ctrl-D<sup>9</sup> 結束 root 的動作回到 登入提示符號。現在請輸入新建立的帳號 penguin 及密碼。<sup>10</sup> 您就會顯示以下的提示命令 列。

<sup>&</sup>lt;sup>6</sup>您可以將 penguin 加入到 adm 群組中以便讀取 /var/log/ 中的紀錄檔。請參閱 passwd (5), group (5), shadow (5), group (5), vipw (8) 和 vigr (8)。對於使用者帳號和群組的官方定義,請參考最近的 Users and Groups (/usr/share/doc/base-passwd/users-and-groups.html)。

<sup>&</sup>lt;sup>7</sup>請在主控台同時按下 Ctrl,Alt(位在鍵盤的左邊) 以及 Delete 鍵。預設的情況是會重新開機。如同'其它主機安 裝提示' 38 說明的,編輯 /etc/inittab 檔案中的 shutdown並加入 -h 就能關機。

<sup>&</sup>lt;sup>8</sup>Debian 系統會這樣做的原因是因為在安裝系統後,良好的檔案權限可以防止非特權用户破壞系統。當然,依 然有許多存在的漏洞被發現,但這些問題應該是留給適合 Securing Debian Manual (http://www.debian.org/ doc/manuals/securing-debian-howto/)的使用者而非閱讀此章節的人。

<sup>&</sup>lt;sup>9</sup>同時按下左Ctrl鍵和D。不需要按Shift即使是控制字元被標示成大寫的 "control D"。

<sup>&</sup>lt;sup>10</sup>如果您將 penquin 換成 root 登入,就會取得 root 權限。如果您要重新取得 root 權限請執行這個動作。

penguin@foo:penguin\$

從現在開始,底下的範皆使用簡化的命令提示列來說明。我將會使用

- #: root 命令提示列
- \$:非特權帳號的命令提示列

我們將以最簡單的方式 'Midnight Commander (MC)' 44 和適當的方法 'Unix-like 工作環境' 46 來學習 Debian 系統。

# 4.2 Midnight Commander (MC)

Midnight Commander (MC)是 Linux 主控台和其它終端環境中的 GNU "瑞士軍刀"。對於新 手來說,選單式的控制會比傳統 Unix 指令好上手多了。

使用這個指令來流覽 Debian 系統。這是最好的方式來學習。請使用方向鍵和 Enter 鍵來流覽 主要的檔案系統。

- /etc 以及子目錄。
- /var/log 以及子目錄。
- /usr/share/doc 以及子目錄。
- /sbin和/bin

## 4.2.1 加強您的 MC

爲了使 MC 在離開時改變工作目錄,您需要修改 ~/.bashrc (或 /etc/bash.bashrc,由.bashrc 呼叫),使用 -P 選項,就像在説明頁面 mc(1) 中的敘述。<sup>11</sup>

#### 4.2.2 啓動 MC

\$ mc

在 MC 中可使用選單來完成所有的檔案操作,只需要使用者小小的嚐試一下就行了。按下 F1 顯示說明視窗。您可以搭配方向鍵和功能鍵來使用 MC。<sup>12</sup>

#### 4.2.3 MC 的檔案管理功能

預設狀態是兩個目錄面板列出檔案列表。另一種好用的模式是將右邊視窗設定為 "information",以查看檔案存取權限等資訊。下面是一些基本的按鍵。如果 gpm daemon 在運作的 話,你還可以使用滑鼠。(在 MC 中進行剪下和貼上操作時別忘了按下 shift 鍵。)

• F1: 説明選單

<sup>11</sup>如果您不清楚我説明的,可以先跳過。

<sup>&</sup>lt;sup>12</sup>如果是在像 kon 和 Kterm for Japanese 等使用某些圖形特性的終端的話,則在命令列中加上 -a 可以預防 MC 出現一些問題。

- F3: 内建的檔案檢視器
- F4: 内建的编辑器
- F9: 啓動下拉選單
- F10: 離開 Midnight Commander
- Tab: 在兩個視窗間移動
- Insert: 標記檔案以方便多檔案的操作, 如複製
- Del: 刪除檔案 (小心—以 MC 設定為安全刪除模式。)
- 方向鍵:就是方向鍵

#### 4.2.4 MC 的命令列技巧

- 任何 cd 命令會改變所選畫面的目錄。
- Ctrl-Enter 或 Alt-Enter 會將檔案名稱複製到命令列上。配合 cp 或 mv 命令使用來做命令 列的編輯。
- Alt-Tab 會顯示出 shell 的檔名擴展以供選擇。
- 可指定 MC 兩個目錄視窗的起始目錄;例如 mc /etc /root。
- Esc + 數字鍵 == Fn (也就是說, Esc + '1' = F1 等等。Esc + '0' = F10)
- Esc 鍵 == Alt 鍵 (= Meta , M-);也就是説,輸入 Esc + 'c' 就等於 Alt-c

#### 4.2.5 MC 的编輯器

内建編輯器的剪下一貼上方式很有意思。按F3標記起始點,再次按F3標記終止點,而中間的 選取部分會高亮度顯示。然後你可以移動游標到某處按下F6,被選取部分就會移到該處。如 果你按下的是F5,選中部分就會複製到該處。F2會儲存檔案,F10是退出,所有方向鍵的作 用則和它們的名字一樣直觀。

可用編輯器直接打開某個檔案:

- \$ mc -e filename\_to\_edit
- \$ mcedit filename\_to\_edit

雖然它不是多視窗編輯器,但你可利用多個 Linux 主控台來達成相同的效果。要在視窗間進行 複製動作,可使用 Alt-Fn 來切換虛擬主控台,然後使用 "File->Insert file" 或 "File->Copy to file" 將一個檔案中的內容複製到另一個檔案。

可指定任何外部编辑器來取代内建的编輯器。

許多程式使用環境變數 EDITOR 或 VISUAL 來決定使用哪個編輯器。如果你用不慣 vim,可在 ~/.bashrc 中加上幾行來指定使用 mcedit :

```
...
export EDITOR=mcedit
export VISUAL=mcedit
...
```

我推薦設定為 vim。習慣 vi(m) 是件好事,因為在 Linux/Unix 世界裡你總找得到它們。<sup>13</sup>

#### 4.2.6 MC 的檢視器

非常聰明的檢視器。它是在文件中搜索詞句的利器。我總是用它在 /usr/share/doc 目錄下 找檔案,用它來瀏覽大量的 Linux 資料是最快的方法。檢視器可以像這樣直接啓動:

\$ mc -v filename\_to\_view

#### 4.2.7 MC 自動啓動特性

在檔案上按 Enter, 會啓動相關的程式來處理該檔案。這是 MC 的一個極方便的特性。

executable: 執行命令 man, html file: 將檔案內容導向 (Pipe) 至檢視軟體 tar, gz, rpm file: 像子目錄般的瀏覽其內容

爲了使這些文件特性生效,不要將可閱讀的檔案設成可執行檔。可使用 MC 的檔案選單或使用 chmod 來改變檔案屬性。

## 4.2.8 MC的FTP虛擬檔案系統

MC 可通過 FTP 來存取網路上的檔案。按 F9 轉到選單列,輸入 'p' 啓動 FTP 虛擬檔案系統。 以 username: passwd@hostname.domainname 格式輸入 URL ,遠端的目錄就會像本地端 目錄一樣顯示出來。

試著流覽 http.us.debian.org/debian 中的 Debian 檔案。請參閱 'Debian archives' 5 了 解這些是如何組織起來的。

# 4.3 Unix-like 工作環境

雖然 MC 能幫助你完成許多事情,但是學會如何從 shell 命令列執行指令並熟習 Unix-like 環境 是非常重要的。<sup>14</sup>

<sup>&</sup>lt;sup>13</sup>事實上,vi或 nvi 是您在那都能取得的。我推薦新手使用 vim 是因為按下 F1 就能看到說明文件且類似 於前者並提供更強大的功能。如果您不熟悉 vim 或不喜歡用,那麼您可繼續使用 mcedit 來處理大部份系統 維護的工作。自從 mcedit是以 8-bit 編碼且傻瓜(指不理會檔案編碼),用它來修改未知編碼的檔案是比較方便 的。mcedit無法正確顯示 UTF-8 的檔案。

 $<sup>^{14}</sup>$ 在本章節中, shell 指的是 bash。想要伸入了解各個 shell 的差異,請參閱 'Shell'  $^{203}$ 。

### 4.3.1 特殊的組合鍵

在 Unix-like 環境中,有許多組合鍵有特別的意義。<sup>15</sup>

- Ctrl-U:清除一行中游標之前的文字。
- Ctrl-H: 移除游標前的一個字元。
- Ctrl-D: 結束輸入。(如果您在 shell 下,就會跳出該 shell)
- Ctrl-C: 終止正在執行的程式。
- Ctrl-Z: 暫時停止程式。(將該工作放到背景,參閱'command &' 53)
- Ctrl-S: 停止輸出到畫面上。(類似 Scorll Lock)<sup>16</sup>
- Ctrl-Q: 回復輸出到畫面上。

預設的 shell 是 bash , 它能記住使用過的命令 (history-editing) 和命令列擴展 (tab-completion) 的功能來輔助互動式的使用。

- up-arrow: 開始收尋過去的命令。
- Ctrl-R: 開始漸進示收尋過去的命令。
- TAB: 擴展要輸入的檔案到命令列上。
- Ctrl-V TAB: 輸入 TAB, 而不是命令列擴展

其他重要的組合鍵:

- Ctrl-Alt-Del: 重新啓動/關閉系統,參閱'其它主機安裝提示' 38。
- Left-click-and-drag mouse: 選取並拷貝至剪貼簿。
- Click middle mouse button: 將剪貼簿內容貼至游標處。
- Meta-key (只針對 Emacs) 是設定在左邊的 Alt。有些系統會使用 Windows-key 當 Meta-key。

為了在 Linux 文字主控台中使用滑鼠,您必須使用 gpm 服務。<sup>17</sup> 參閱 '滑鼠設定' 32。

## 4.3.2 基本的 Unix 指令

我們來學一些基本的 Unix 指令吧。<sup>18</sup> 試著由非特權帳號 penguin來執行下列的指令:

- pwd
  - 顯示目前的工作目錄。
- whoami
  - 顯示目前的使用者
- file foo
  - 顯示foo檔案的型別。
- type -p commandname
  - 顯示執行檔commandname的檔案路徑。
  - which *commandname*也是相同功能。<sup>19</sup>
- type commandname
  - 顯示執行檔 commandname 的資訊。

<sup>15</sup>在正常的 Linux 文字主控台下,只有左邊的 Ctrl 和 Alt 是有用的。

<sup>16</sup>您可以利用 stty(1) 來關閉這個終端機功能。

<sup>17</sup>在 X 系統環境下, 滑鼠的功能跟 Xterm 軟體一樣。 In the X Window environment, the mouse functions in the same way with the Xterm program.

<sup>18</sup>我在這邊使用 "Unix" 當作通解。任何的 Unix 系列的作業系統通常有提供一樣的指令。Debian 也不例外。別 擔心,如果有些指令不能運作,這些範例將不會在這邊出現。

<sup>19</sup>如果在 shell 使用 alias, 則輸出的結果就不相同。

```
• apropos key-word
   - 找出與 key-word 有關的說明文件。
   - man -k key-word也具有相同功能。
• whatis commandname
   - 顯示指令commandname的單行敘述。
• man -a commandname
   - 顯示指令 commandname的說明 (Unix 風格)
• info commandname
   - 顯示指令 commandname 較豐富的說明 (GNU 風格)
• ls
   - 列出目錄的內容。(非點開頭的檔案和目錄)<sup>20</sup>
• ls -a
   - 列出目錄的内容。(所有的檔案和目錄)
• ls -A
   – 列出目錄的內容。(幾乎所有的檔案和目錄,也就是略過".."和".")
• ls -la
   - 列出目錄下所有的檔案和細節資訊。參閱 'Debian 的檔案系統' 58。
• ls -d
   - 列出目前目錄下所有的目錄。
• lsof foo
   - 列出以開檔的 foo 狀態。
• mkdir foo
   - 在目前的目錄下產生新目錄: foo。
• rmdir foo
   - 在目前的目錄下移除目錄: foo。
• cd fon
   - 切換目錄到目前目錄下的 foo 或是CDPATH變數所定義的目錄下的目錄。參閱文
    件builtins(7)中的cd。
• cd /
   - 切換到根目錄。
• cd
   - 切換到使用者的家目錄。
• cd / foo
   - 切換到絕對路徑為/foo的目錄。
• cd ..
   - 切換到上一層目錄。
• cd ~foo
   - 切換到家目錄中的 foo 目錄。
• cd -
   - 切换到先前的目錄。
• </etc/motd pager
   - 使用預設的 pager 顯示 /etc/motd 的内容。參閱 'command < foo' 54。<sup>21</sup>
```

• touch junkfile

<sup>&</sup>lt;sup>20</sup>Unix 有個傳統是將檔案名稱以"."開頭爲隱藏檔。這些檔案通常是紀錄設定資訊和使用者偏好設定。

<sup>&</sup>lt;sup>21</sup>全新精簡的 Debian 系統預設的 pager 爲 more,但不能上下捲動。執行 apt-get install less 來安裝 less 就會變成預設的 pager 且您可以使用方向鍵來捲動上下頁。

- 建立一個空檔案: junkfile。 • cp foo bar - 複製一已存在的檔案: foo 並命名成:bar。 • rm junkfile - 移除 junkfile 檔案。 • mv foo bar - 更名一已存在的 foo 檔案成 bar 檔名。 • mv foo bar/baz - 移動一已存在的 foo 檔案到新的位置且更名成 bar/baz。bar 目錄必須存在。 • chmod 600 foo - 設定一已存在的檔案: foo 的權限為其他帳號皆不可讀和寫。(任何人皆不可執行) • chmod 644 *foo* - 設定一已存在的檔案: foo 的權限為其他帳號皆可讀但不可寫入。(任何人皆不可執 行) • chmod 755 *foo* - 定一已存在的檔案: foo 的權限為其他帳號皆可讀但不可寫入。(任何人皆可執行) • top - 全螢幕顯示程序的資料。輸入"q"終止。 • ps aux | pager - 使用 BSD 風格顯示全程序的資料。參閱 'command1 | command2' 53。 • ps -ef | pager - 使用 Unix system-V 風格顯示全程序的資料。 • ps aux | grep -e "[e]xim4\*" - 顯示全部 exim 或 exim4 程序。執行 man grep 來閱讀 grep(1) 學習正規式表示 法。22 • ps axf | pager - 使用 ASCII art 輸出顯示全程序的資料。 • kill 1234 - 砍掉 pid 為 1234 的程序。參閱 '中止一個進程' 100。 • grep -e "pattern" \*.html - 收尋並顯示目錄下所有副檔名為.html 的"原型"。 • gzip foo - 使用 Lempel-Ziv coding (LZ77) 壓縮 foo.gz。 • qunzip foo.qz - 解壓縮 foo.gz 產生 foo。 • bzip2 foo - 使用 Burrows-Wheeler block sorting text compression algorithm 和 Huffman coding (比 gzip 壓縮率更高) 來壓縮 foo 成 foo.bz2 • bunzip2 foo.bz2 - 解壓縮 foo.bz2 產生 foo。 • tar -xvvf foo.tar

- 解開 foo.tar 保存檔。
- tar -xvvzf *foo*.tar.gz

<sup>&</sup>lt;sup>22</sup>正規式表示法中的 [和] 可使 grep 避免收尋到自己。至於表示法中的 4\* 代表符合四個字元的次數大於零來 使grep匹配exim和exim4。雖然 \*皆可用在 shell filename wild card 和正規表示法,但兩者的意義是不同的。

- 解開 gzip 壓縮檔: foo.tar.gz。
- tar -xvvf --bzip2 *foo.tar.bz2* 
  - 解開 bzip2 壓縮檔:foo.tar.bz2。<sup>23</sup>
- tar -cvvf foo.tar bar/
  - 打包目錄 bar/ 成 foo.tar 保存檔。
- tar -cvvzf foo.tar.gz bar/

   - 壓縮和打包目錄 bar/到 foo.tar.gz 保存檔。
- tar -cvvf --bzip2 foo.tar.bz2 bar/
  - 壓縮和打包目錄 bar/ 到 foo.tar.bz2 保存檔。24
- zcat README.gz | pager
   使用預設的 pager 來顯示壓縮檔 README.gz 的内容。
- zcat README.gz > foo
  - 建立 foo 檔案而内容為已解壓縮的 README.gz。
- zcat README.gz >> foo

   附加已解壓縮的 README.gz的內容到 foo 檔案。(如果該檔案不存在,請先建立。)
- find . -name pattern
   使用 shell 原型 收尋相符的檔名。(速度較慢)
- locate -d . *pattern* 使用 shell 原型 收尋相符的檔名。(使用定期產生的資料庫,較快)

請先使用上述的指令當作測試來探勘系統目錄。如果您有任何指令上的問題,請閱讀說明文 件。比如說,這些指令是好的出發點:

- \$ man man
  \$ man bash
- \$ man ls

現在使用 vim 的好機會,請按下 F1。您應該會看到至少前 35 行。然後實作線上測驗,將游標 移至 |tutor | 並按下 Ctrl-]。參閱 '編輯器' 189 了解更多編輯器。

請注意,許多從 GNU 和 BSD 來的 Unix-like 指令皆有簡潔的説明,您可以使用以下其一的方法叫出來(有些情況是不用加任何參數):

- \$ commandname --help
- \$ commandname -h

也順便練習一下'Debian技巧' 93的範例來練功吧。

## 4.3.3 The command execution

相信您現在比較清楚如何使用 Debian 系統了。我們來探勘 Debian 中更進階的指令技巧。25

<sup>&</sup>lt;sup>23</sup>在這裡用 --bzip2 來替代較短的 -j 來保證 Potato 中舊版的 tar 能正常工作。

<sup>&</sup>lt;sup>24</sup>同樣地,--bzip2是確保相同性。

<sup>&</sup>lt;sup>25</sup>在這裡,我省略了一些東西,這對於新手比較好入手。詳細說明請參閱 See bash (1)。

#### 4.3.4 簡單的命令

一個簡單的指令是以下的排列

- 1 變數宣告 (額外的)
- 2 指令名稱
- 3 參數 (額外的)
- 4 導向(額外的:>,>>,<,<<...等)
- 5 控制運算子 (額外的: & & , | | ;<newline> , ; , & , (,))

對於置換和引用,參閱'命令列操作'207。

#### 4.3.5 指令執行和環境變數

典型的指令執行是使用下述的 shell 指令列: 26

\$ date
Sun Oct 26 08:17:20 CET 2003
\$ LC\_ALL=fr\_FR date
dim oct 26 08:17:39 CET 2003

在這裡,date是前景工作。至於環境變數 LC\_ALL 是指:

• 第一個指令未設定(系統預設是跟 C 一樣) LC\_ALL

• 第二個指令將 LC\_ALL 設成 fr\_FR (French locale)

大部份的指令在執行時不會比環境變數更早執行。對於上面的範例來說,你也可以這樣做:

\$ LC\_ALL=fr\_FR
\$ date
dim oct 26 08:17:39 CET 2003

當你看到這裡時,輸出的結果是受到環境變數的影響而產生 French 的文字。如果您想要環境 變數也影響到子程序(舉例來說,當使用 shell script),您必須 "export" 它:

\$ export LC\_ALL

#### 4.3.6 指令收尋路徑

當您在 shell 輸入指令時, shell 會 PATH 這個環境變數中列舉的目錄來找指令。PATH 同時也是 shell 的收尋路徑。

在預設安裝的 Debian 下,使用者的PATH 不會包含/sbin/。所以當您需要從 /sbin/執行 一些像 ifconfig的指令時,您必須修改 PATH 的值並包含該路徑。PATH通常是設定在 ~ /.bash\_profile 之中,參閱 'Bash 設定' 31。

<sup>&</sup>lt;sup>26</sup>要完成下列的輸出,您必須安裝 French locale,參閱 'Locales' 151。這並不是該指南的基本。這只是爲了說明 它另一個結果。

#### 4.3.7 命令列参數

部份的指令需要額外的參數才能執行,這些參數通常是以 - 或 -- 開頭並透過指令來呼叫。

```
$ date
Mon Oct 27 23:02:09 CET 2003
$ date -R
Mon, 27 Oct 2003 23:02:40 +0100
```

這裡的參數 -R 會改變 date 的輸出結果而變成 RFC-2822 相容的字串。

## 4.3.8 Shell wildcards

有時候您希望能同時處理相同類別的檔案但不想一個一個輸入時, shell wildcards的"檔名原型擴展"就能解決這樣的問題。

- \*

  符合任何0個或更多個的字元。
  這不符合檔名以"."開頭的檔案。

  ?

  只符合一個字元。

  [...]

  只符合括號中的一個字元。
- [a-z] - 只符合a 和 z 中的一個字元。
- [^...]
   符合括號中沒有列舉的一個字元("^"並不包含)。

舉例來說,思考一下底下的範例:

```
$ mkdir junk; cd junk; touch 1.txt 2.txt 3.c 4.h .5.txt
$ echo *.txt
1.txt 2.txt
$ echo *
1.txt 2.txt 3.c 4.h
$ echo *.[hc]
3.c 4.h
$ echo .*
. .. .5.txt
$ echo .[^.]*
.5.txt
$ echo [^1-3]*
4.h
$ cd ..; rm -rf junk
```

#### 4.3.9 指令的傳回值

每個指令會回傳 exit 狀態當成回傳值。

• 如果指令成功執行完成的話,則回傳值=0。

如果指令因錯誤終止,則回傳值不為()。

執行後的回傳值可經由 \$? 環境變數來存取。

```
$ [ 1 = 1 ] ; echo $?
0
$ [ 1 = 2 ] ; echo $?
1
```

請注意, shell 使用邏輯計算回傳值時, success的話就當成邏輯 TRUE。這並不是那麼直覺, 因為success有代表 zero 的意思。

參閱 'Shell 條件式' 206。

## 4.3.10 標準的指令順序

我們試著記憶底下的 shell 指令用法。閱讀過這些用法後也請參考 'Shell 參數' 205, 'Shell 重導 (redirection)' 205, 'Shell 條件式' 206 和 '命令列操作' 207。

#### command &

該 command 指令是在背景中的子 shell 執行。背景的工作允許使用者在一個 shell 中執行多個 程式。

背景程序的管理包含了 shell 本身的 jobs, fg, bg 和 kill。請閱讀 bash(1) 中的 "SIG-NALS", "JOB CONTROL" 和 "SHELL BUILTIN COMMANDS"。<sup>27</sup>

### command1 | command2

command1 的 standard output 是餵給 command2 的stand input。兩個指令可以同時執行。這就叫作 pipeline。

command1 ; command2

command1 和 command2 是按照順序地執行。

#### command1 && command2

command1 先被執行。如果成功的話,則command2也會依序的執行。command1 和 command2皆成功執行,則傳回 success。

<sup>27</sup>Debian 系統是多工的系統。

#### command1 || command2

command1 先被執行。如果不成功的話,則command2還是會依序的執行。command1 或 command2其一成功執行的話,則傳回 success。

#### command > foo

將command的結果重導到foo檔案。(會覆寫)

command >> foo

將command的結果重導到foo檔案。(會附加)

#### command > foo 2>&1

將command的 standard output 和 standard error 重導到 foo檔案。

#### command < foo

將command的 standard input 導到 foo檔案。試試看以下的例子:

```
$ </etc/motd pager
... (the greetings)
$ pager </etc/motd
... (the greetings)
$ pager /etc/motd
... (the greetings)
$ cat /etc/motd | pager
... (the greetings)
```

雖然四個用法皆顯示同樣的結果,但最後的例子會再跑一次 cat 指令,這會多浪費一些資源。

#### 4.3.11 指令別名

您可以爲常用的指令設定別名。例如:

\$ alias la='ls -la'

現在, la 會以長清單格式來列出所有的檔案,也就是 ls -la 的縮寫。 您可以使用 type 指令來辨識真正的路徑。例如:
```
$ type ls
ls is hashed (/bin/ls)
$ type la
la is aliased to `ls -la'
$ type echo
echo is a shell builtin
$ type file
file is /usr/bin/file
```

這裡的 1s 是最近被查詢到的,然而 file 並沒有,因此 1s 是 "hashed",也就是說 shell 有一個紀錄可以快速找到 1s 的位置。

#### 4.4 Unix-like 文字處理

- 在 Unix-like 系統上有一些常用的標準文字處理工具。
  - 不支援正規表示法的:
    - head 輸出檔案的第一行。
    - tail 輸出檔案的最後一行。
    - sort 排序文字檔行列。
    - uniq 删除已排序的檔案中重複的資料。
    - tr 轉譯或刪除文字。
    - diff 行與行的檔案比較。
  - 支援基本的正規表示法 (BRE):
    - grep 依照收尋原型來收尋文字。
    - ed 是個最原始基本的編輯器。
    - sed 是串流資料編輯器。
    - vi 是全螢幕編輯器。
    - emacs 是全螢幕編輯器。
  - 支援擴充的正規表示法(ERE):
    - egrep 依照收尋原型來收尋文字。
    - awk 支援簡單的文字處理。參閱 'Awk' 207。
    - perl 幾乎所有的文字處理皆能達成。參閱 'Perl' 209。

參閱 'Regular-expression substitution' 105, '精巧的管道指令輔助script' 108 和 '輕巧的 Perl Script' 109 取得更多 script 範例。

#### 4.4.1 正規表示法

正規表示法是用在許多文字處理的工具上。有點類似 shell wildcards (參閱 'Shell wildcards' 52),但卻更複雜與強大。

正規表示法可用文字或metacharacters來描述比對原型。metacharacter 是具有特殊意義的字元。有兩種主要的類別:BRE 和 ERE,端看您使用的文字工具('Unix-like 文字處理' 55)決定。

對 EREs 來説, metacharacters 包含"\ . [] ^ \$ \* + ? () { } |"。這個表示法意 思是: • C - 符合非 metacharacter "c"。 • \c - 符合字母"c"。 • . - 符合包含换行的任何字元。 • ^ - 符合字串的第一個字元。 • \$ - 符合字串的最後一個字元。 • \< - 符合文字的開頭。 • \> - 符合文字的結尾。 • [abc...] - 這個字元列表符合 "abc..." 中任何一個字元。 • [^abc...] - 這個字元列表符合非 "abc..." 中任何一個字元。 • r\* - 符合"r"這個敘述 0 次以上的文字。 • r+ - 符合"r" 這個敘述1次以上的文字。 • r? - 符合"r" 這個敘述 () 或 1 次的文字。 • r1|r2 - 符合"r1"或"r2" 敍述之一的文字。 • (r1|r2)

- 符合"r1"或"r2" 敘述之一的文字並以 bracketed 表示法處理。

在 BREs 裡, metacharacters: "+ ? () { } |" 是無作用的,取而代之的是倒斜線的"\+ \? \( \) \{ \} \|"。所以説,使用 BREs 時,(r1|r2) 必須像\(r1|r2\) 用倒斜線隔 開。雖然 emacs 也支援 BRE,但改用 "+ ?" 來當作metacharacters。所以不需要使用雙引號 來隔開。參閱 '置換表示法' 56 了解這些metacharacters組合怎麼使用。

舉例來說, grep 可以使用正規表示法來收尋文字:

\$ egrep 'GNU.\*LICENSE|Yoyodyne' /usr/share/common-licenses/GPL GNU GENERAL PUBLIC LICENSE GNU GENERAL PUBLIC LICENSE Yoyodyne, Inc., hereby disclaims all copyright interest in the program

#### 4.4.2 置换表示法

在置换表示法中,底下的字元有特別的意義:

• &

- 顯示符合正規表示法的字串。(emacs下請用 \&)

如要使用 Perl 置換字串,請改用 \$n 代替\n和&。因爲後兩各在 Perl 中是無意義的。 舉例說明:

```
$ echo zzzlabc2efg3hij4 | \
    sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/=&=/'
zzz=labc2efg3hij4=
$ echo zzzlabc2efg3hij4 | \
    sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/\2===\1/'
zzzefg3hij4===labc
$ echo zzzlabc2efg3hij4 | \
    perl -pe 's/(1[a-z]*)[0-9]*(.*)$/$2===$1/'
zzzefg3hij4===labc
$ echo zzzlabc2efg3hij4 | \
    perl -pe 's/(1[a-z]*)[0-9]*(.*)$/=&=/'
zzz=&=
```

請特別注意各處理工具在置換時所使用的**分隔**符號,以及軟體如何用正規表示法比對這些文字 的方式。

在編輯器中使用這些表示法也可以用來移動游標和置換文字。

請閱讀以上這些指令相關的說明文件。

## 4.5 Unix-like 檔案系統

在 GNU/Linux 和其他 Unix-like 的系統上,檔案是有組織地放置在目錄下。<sup>28</sup> 全部的檔案和 目錄被整理放在一個階乘和樹狀的根目錄(/)。

這些檔案和目錄可以放在不同的硬體裝置上。mount (8) 指令能掛載硬體裝置並加入到這巨大的檔案樹系統上。反過來說,umount (8)則可以卸載這些裝置。

#### 4.5.1 Unix 檔案基礎

以下是一些基礎觀念:

- 檔案是大小寫相異。也就是說 MYFILE 和 MyFile是不同的檔案
- 根目錄以 / 表示。可別把 "root" 跟 root 帳號搞混了喔。參閱 '用 root 登入 shell 提示符號' 41。

<sup>&</sup>lt;sup>28</sup>在某些系統上,目錄(diretory)也被稱為 資料夾(folder)。

- 每個目錄皆可用字母或符號來命名,但就是不能用/。<sup>29</sup>根目錄是一個例外,它的檔 名:/(叫做"斜線"或"根目錄")且不能更名。
- 每個檔案或目錄皆設計成 fully-qualified filename, absolute filename或path,必須有順序地輸入路徑才能切換到最後的目錄。以上三者是同等的意思。全部的絕對路徑以/開頭且每個檔案或目錄的路徑都包含/符號。第一個/是目錄的名稱,其餘的皆爲分隔符號以便區分檔名的每個部份。以上的說明容易造成混淆。請看一下底下的範例:

/usr/share/keytables/us.map.gz 這是一個完整的檔案名稱,有些人也稱作 **path**。但大部分都只用 us.map.gz 來當作檔 名。 30

- root 目錄有幾個分支,例如 /etc/ and /usr/。這些子目錄底下有更多的子目錄,例如 /etc/init.d/ 和 /usr/local/。總括來說,這就叫做 目錄樹(directory tree)。您可以想像一個絕對路徑是從該樹的根(/)延伸到最後的分支(檔案)。您也曾聽過別人稱爲家 族樹,所以每個子目錄都有其父母或祖先,而路徑就是檔案完整的起源。至於相對路徑 並不是以 root(/) 目錄開頭。您必須牢記.. / 代表上一層的目錄。
- 至於硬體裝置是沒有任何目錄對應的。這跟 CP/M, DOS, 和 Windows 中的所有檔案是以 硬體裝置(例如, C:\)開頭的檔名不同。請參閱'Debian 的檔案系統' 58。

有興趣的話可以參閱Filesystem Hierarchy Standard (/usr/share/doc/debian-policy/fhs/fhs.txt.gz),該文件詳細説明了階層式檔案系統的細節。您要記住的是底下這些目錄的用途:

- /
  - / 代表 root 目錄。
- /etc/
   所有的系統設定檔案皆放在該目錄下。
- /var/log/
  - 所有的系統日誌檔皆放在該目錄下。
- /home/
   該目錄包含了非特權帳號的所有家目錄。

#### 4.5.2 Debian 的檔案系統

同於 Unix, Debian 系統提供了一個檔案系統,能夠統一化所有硬碟,儲存媒體,主控台或透過串列埠遠端存取硬體的動作。

Debian 系統中每個檔案,目錄,named pipe 或硬體裝置都有一個inode 的資料結構,記載該 檔案的相關屬性,例如:使用者所擁有 (擁有者),屬於那一個群組,最後存取的時間等。參閱 /usr/include/linux/fs.h 了解 Debian GNU/Linux 系統中的 inode結構的意義。

將硬體統一化成檔案的表示法是非常有用的,這允許我們可以使用相同的方法來存取完全不同的硬體。

全部的檔案可以存放在一個硬碟,或者 20 個硬碟,有些硬碟甚至是分散在網路上不同的電

<sup>&</sup>lt;sup>29</sup>您**可以**用任意的字母或符號來命名檔案,但實際上這不是好的命名方式。最好能避免使用一些特殊意義的字元,比如 spaces, tabs, newlines 和其他特殊意義的字元:{}()[]'、"\/><|;! # & ^ \* % @ \$。 如果您想要分隔檔名中的字串,逗號(,),連字符號(-)和底線(\_)是比較推薦的。您也可以用大寫,像是這樣:LikeThis。

<sup>&</sup>lt;sup>30</sup>關於 path 也有不同的用法。參閱 '指令收尋路徑' 51。透過上述的解釋應該就可了解。

腦。您無法想像在查詢檔案或目錄時,所有的指令都能正常運作且不用理會真正的檔案是放在 哪一個實體裝置上。

#### 4.5.3 檔案和目錄的存取權限

檔案和目錄的存取權限對下面三種使用者分別定義:

- 檔案擁有者 user (u) ,
- 檔案所屬群組的使用者 group (g) 和
- 所有使用者 **other** (o) 。

對檔案來說,每個權限分別對應到一些可做的動作:

- 讀取read (r): 查看檔案的内容
- 寫入write (w):對檔案作修改
- 執行execute (x):把檔案當作指令來執行

對目錄來說,每個權限分別對應到一些可做的動作:

- 讀取read (r):列出目錄的内容
- 寫入write (w):在目錄中加入或刪除檔案
- 執行execute (x):存取目錄中的檔案

在這裡, 目錄的執行權限不只代表允許讀取目錄內的檔案,也代表允許看到它們的屬性,像是 檔案大小和更動時間等。

使用 1s 來顯示檔案和目錄的權限資訊 (和更多其他的資訊). 請參閱 1s(1)。當使用 1s 的 -1 選項,則會依下列順序來顯示相關資訊:

- 檔案型態 (第一字元)
  - -: 一般檔案
  - d: 目錄
  - 1: 符號連結
  - c: 字元狀置節點
  - b: 區塊狀置節點
  - p: 命名的管線
  - s: 通訊端
- 檔案的存取權限(之後九個字元,依序是擁有者,群組,其他,各三個字元)
- 檔案的硬連結個數
- 檔案擁有者名稱
- 檔案所屬群組名稱
- 檔案大小 (以 bytes 為單位)
- 檔案的目期和時間 (mtime)
- 檔案的名稱。

在 root 帳號下可使用 chown 改變檔案的擁有者。要改變檔案的所屬群組,可以用檔案擁有者 或 root 的身份來執行 chgrp。要改變目錄的存取權限,可以用檔案擁有者或 root 的身份來執 行 chmod。控制foo檔案的語法是:

- # chown newowner foo
- # chgrp newgroup foo
- # chmod [ugoa][+-=][rwx][,...] foo

請參閱 chown (1), chgrp (1) 和 chmod (1) 以取得更多資訊。

舉例來說,如果要改變目錄的擁有者為 foo 以及群組為 bar,請用 root 帳號執行以下指令:

- # cd /some/location/
- # chown -R foo:bar .
- # chmod -R ug+rwX,o=rX .

有3個更特殊的權限位元:

- set user ID (s 或 S 取代擁有者的 x),
- set group ID (s 或 S 取代群組的 x) 和
- sticky bit (t 或 T 取代其他使用者的 x)。

在這裡,如果設定這些特殊權限設定後蓋住了原本顯示的執行權限的話,則1s -1 的輸出就 會用大寫來表示。

在可執行檔上設定 set user ID 的話,會以這個檔案擁有者的身份 (如 root) 來執行這個可執行 檔。同樣地,在可執行檔上設定 set group ID 的話,會以這個檔案的群組身份 (如 root) 來執行 這個可執行檔。因爲這些設定會導致安全性上的問題,要打開它們的話要格外小心。

在目錄上設定 set group ID 的話,會使用類似 BSD 的檔案建立規則,也就是在此目錄中建立的檔案都會屬於此目錄的群組。

在目錄上設定e sticky bit 可以防止非檔案擁有者來刪除此目錄中的檔案。在一些大家可以寫入的目錄,如 /tmp 或是在允許群組寫入的目錄下,爲了要保持檔案內容的安全,不只要關上檔案的寫入權限,還要在目錄上設定 sticky bit。不然的話,任何擁有目錄寫入權限的人都可以 刪除檔案,並且建立一個新的同名檔案。

這裡有些關於檔案權限的有趣例子。

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/pppd
crw-rw---- 1 root
                       dip
                                108, 0 Jan 18 13:32 /dev/ppp
                                    1051 Jan 26 08:29 /etc/passwd
-rw-r--r--
             1 root
                        root
                                     746 Jan 26 08:29 /etc/shadow
-rw-r----
             1 root
                        shadow
-rwsr-xr--
                                  234504 Nov 24 03:58 /usr/sbin/pppd
             1 root
                       dip
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt
            4 root
                       root
                                    4096 Feb 9 16:35 /tmp
                                    4096 Jan 18 13:31 /usr/local
drwxrwsr-x 10 root
                        staff
                                    4096 Jan 19 08:36 /usr/src
drwxrwsr-x
            3 root
                        src
            2 root
                                    4096 Feb 2 22:19 /var/mail
drwxrwsr-x
                       mail
                                    4096 Jan 25 02:48 /var/tmp
drwxrwxrwt
             3 root
                       root
```

另外的辦法是使用數字並搭配 chmod(1) 來設定檔案權限。數字模式使用 3 到 4 數字寬的八進 制數字。每個數字皆有其代表意義:

- 第一個選擇性數字: set user ID (=4), set group ID (=2) 和 sticky bit (=1) 的總合。
- 第二個數字:針對 user 的 read (=4), write (=2) 和 execute (=1) 權限的總合。
- 第三個數字:同上,但針對於 group。
- 第四個數字:同上,但針對 other。

這聽起來很複雜,但其實是相當簡單的。如果您觀察1s -1的結果就能發現前面幾行(2-10)是 代表檔案權限(請用二進制,"-"表示"0","rwx"表示"1")。這些數字如果用八進制來代表檔案 權限的話,相信您一定能夠了解。<sup>31</sup>舉個例子,比如:

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
$ ls -1 foo bar
-rw-r--r-- 1 penguin penguin 0 Nov 3 23:30 foo
-rw-r--r-- 1 penguin penguin 0 Nov 3 23:30 bar
```

預設的檔案權限遮罩可使用 shell 内建的 umask 來設定。參閱builtins (7)。

#### 4.5.4 時間戳記

在 GNU/Linux 檔案中, 有三個時間戳記:

- mtime:修改時間(1s -1),
- ctime: 狀態改變時間 (1s -1c) 和
- atime:最後存取時間(ls -lu)。
- 注意, ctime 並不是檔案建立時間。
  - 覆寫檔案會改變檔案的 mtime , ctime 和 atime 全部的時間戳記。
  - 改變檔案權限或擁有者的話,會改變檔案的 ctime 和 atime 時間戳記。
  - 讀取檔案會改變檔案的 atime 時間戳記。

要注意的是,在 GNU/Linux 系統中,即使只是簡單的讀取檔案,通常也會爲了要更新 inode 中的 atime 而導致寫入的動作。如果以 noatime 選項來掛上檔案系統的話,就可以略過這個 動作而使讀取的動作加快。請參閱 mount (8)。

使用 touch (1) 中的指令來改變檔案的時間戳記。

#### 4.5.5 連結

有兩種方法可以把一個叫 foo 的檔案關連到另一個檔名 bar。

- 硬式連結 (hard link) 是現存檔案的一個別名 (1n foo bar),
- 符號連結 (symbolic link),或是"symlink",是一個特殊的檔案,它指向另一個檔案的 名稱 (ln -s foo bar)。

下面的例子顯示了在使用 rm 指令後,連結數的改變和一些微妙的差異。

\$ echo "Original Content" > foo
\$ ls -1 foo
-rw-r--r-- 1 osamu osamu 4 Feb 9 22:26 foo
\$ ln foo bar # 硬式連結
\$ ln -s foo baz # 符號連結

31當然,這各方法只適用於3個數字寬的表示法。

```
$ ls -l foo bar baz
-rw-r--r--
            2 osamu
                        osamu
lrwxrwxrwx
            1 osamu
                       osamu
-rw-r--r--
            2 osamu
                        osamu
$ rm foo
$ echo "New Content" > foo
$ cat bar
Original Content
$ cat baz
New Content
```

4 Feb 9 22:26 bar 3 Feb 9 22:28 baz -> foo 4 Feb 9 22:26 foo

就像上面的例子所顯示的,符號連結的一般檔案存取權限為"rwxrwxrwx",而實際的存取權限均由它們所指向的檔案來決定。

. 目錄是連結到其所出現的目錄,所以任何新目錄的連結數都是由2起跳的。而.. 目錄是連 結到其所上層目錄,所以目錄的連結數會隨著子目錄數目而增加。

#### 4.5.6 Named pipes (FIFOs)

named pipe 是類似管線的檔案。您可以輸入資料到這個檔案,然後他會輸出到別的地方。因此他也就作 FIFO(First-In-First-Out):第一個送入管線的資料也是第一個送出管線的資料。

如果寫入 named pipe 時,寫入的程序並不會中斷直到該管線讀入爲止。如果讀取 named pipe,讀取的程序會一直等待直到資料讀取完畢才停止。管線的大小永遠是零,因爲它並不儲 存資料,它主要是連結兩個程序,就像 shell 的 |。但是,自從管線被命名之後,兩個程序不需 要在同一行執行或同一個使用者執行。

試著執行以下的例子:

```
$ cd; mkfifo mypipe
$ echo "hello" >mypipe & # put into background
[1] 5952
$ ls -1 mypipe
prw-r--r-- 1 penguin penguin 0 2003-11-06 23:18 mypipe
$ cat mypipe
hello
[1]+ Done echo hello >mypipe
$ ls mypipe
prw-r--r-- 1 penguin penguin 0 2003-11-06 23:20 mypipe
$ rm mypipe
```

#### 4.5.7 Sockets

socket 有點類似 named pipe (FIFO) 且允許程序與程序之間可以交換資料。針對 socket 來 說,這些程序不需要在同一個時間執行或者同一個父程序產生的子程序。這是 inter process communication 的 ENDPOINT。網路上不同的主機透過 socket 也能夠互換資料。

#### 4.5.8 Device files

任何的硬體或虛擬裝置,例如硬碟,顯示卡,螢幕或鍵盤等,皆視爲裝置檔案。/dev/console,主控台是最常見的虛擬裝置。

裝置有兩種型別:

- character device
  - 一次存取一個字元。就是說存取該裝置最小的資料單位元是字元(byte)。
- block 裝置
  - 一次存取較大單位的裝置稱為 block。硬碟就是一個 block 裝置。

您可以讀寫裝置檔案,即使是這些檔案可能採用二進制而讓人難以理解。直接寫入資料到這些 檔案有時候是測試硬體是否已經連線的好方法。例如,您可以列印文字檔到印表機裝置:/dev /1p0或送指令到連接數據機的串列埠上:/dev/ttyS0。但請小心操作,也許會造成重大的損 壞。所以要特別小心。

#### /dev/null 類別

/dev/null 是一個特別的裝置檔案,它會捨棄所有寫入的資料。如果您不需要某些資料,直接丢到 /dev/null。它基本上就是個黑洞。如果您讀取/dev/null,您會馬上得到 end-of-file (EOF) 字元。

/dev/zero也是一樣,差別在於讀取時會得到\0字元(不是數字0喔)。參閱'空檔案'115。

#### 装置節點編號

執行1s您就可以得到裝置節點的號碼:

\$ ls -l /dev	/hda /dev/t	tyS0 /dev/z	ero				
brw-rw	1 root	disk	З,	0 Mar	14	2002	/dev/hda
crw-rw	1 root	dialout	4,	64 Nov	15	09:51	/dev/ttySC
crw-rw-rw-	1 root	root	1,	5 Aug	31	03:03	/dev/zero

請看底下説明:

- /dev/hda 的主要裝置號碼:3,次要號碼是:0。只有disk群組中的使用者可以讀寫 它。
- /dev/ttyS0 的主要裝置號碼:4,次要號碼是:64。只有dialout群組中的使用者可以 讀寫它。
- /dev/zero 的主要裝置號碼:1,次要號碼是:5。任何帳號皆可讀寫它。

在舊式的系統上,安裝過程會使用 /sbin/MAKEDEV 建立裝置節點。參閱 MAKEDEV(8)。

較新的系統上,/dev下的檔案系統是經由類似/proc的檔案系統自動產生。

#### 4.5.9 /proc 檔案系統

/proc 檔案系統是虛擬的檔案系統,紀錄著系統資訊和執行中的程式。

許多人常常因爲 /proc/kcore 太大而傷腦筋。該檔案的大小跟您電腦安裝的記憶體一樣大。 它主要用來進行核心偵錯,但本身並不存在任何地方,所以請別擔心它的容量大小。

參閱 '透過 proc 檔案系統調整核心' 90 和 proc (5)。

## 4.6 X Window 系統

請參閱 'X' 128。

## 4.6.1 啓動 X Window System

X Window 系統會跟著 xdm-like 等圖形登入軟體執行時啓動,或者在主控台輸入:

\$ exec startx

#### 4.6.2 X Window 系統的選單功能

自從 X 環境提供了許多的視窗管理器之後,他們的使用者介面都有些許的差異。請牢記按下右 鍵會帶出系統選單。該功能一直都有。

- 要取得 shell 命令列,點選選單中的 Xterm:
  "XShells" -> "XTerm".
- 至於圖形化的網頁瀏覽器,點選選單中的 Mozilla:
   "Apps" -> "Net" -> "Mozilla Navigator".
- 至於圖形化的 PDF 瀏覽器,點選選單中的 Xpdf:
   "Apps" -> "Viewers" -> "Xpdf".

如果您找不到 menu, 請安裝相關的套件。參閱 / 體驗 Debian 套件管理 / 72。

#### 4.6.3 X Window 系統的鍵盤順序

底下是執行 X Window 系統時重要的組合控制鍵。

- Ctrl-Alt-F1 到 F6: 切換到其他的虛擬終端機 (從 X window, DOSEMU 等)
- Alt-F7: 返回 X window 畫面
- Ctrl-Alt-minus: 降低 X window 的解析度 (這裡的 minus(-) 是指數字鍵上的減號)
- Ctrl-Alt-plus: 提高 X window 的解析度 (這裡的 plus(+) 是指數字鍵上的加減號)
- Ctrl-Alt-Backspace: 終止 X Server
- Alt-X, Alt-C, Alt-V: 同於 Windows/Mac 上的剪下, 複製, 貼上。在某些程式如 Netscape Composer 需改用 Alt 來代替 Ctrl。

## 4.7 後續的學習指南

現在,我推薦您閱讀 The Linux Documentation Project: Guides (http://www.tldp.org/guides.html) 網站上主要的引導書籍:

- "The Linux System Administrators' Guide",
  - 此書包含了如何讓系統永不停機,使用者帳號管理,備份,系統設定等觀念。
  - 套件:sysadmin-guide
  - 檔案: /usr/share/doc/sysadmin-guide/html/index.html
  - 網址:http://www.tldp.org/LDP/sag/index.html
- "The Linux Network Administrator's Guide, Second Edition",
  - 這是一本駕馭 Linux 網路管理的參考手册。
  - 套件: (not available)
  - 檔案: (notapplicable)
  - 網址: http://www.tldp.org/LDP/nag2/index.html
- *Linux: Rute User's Tutorial and Exposition* 
  - 管理 GNU/Linux 系統的精裝版線上書籍。
  - 作者: Paul Sheer
  - 由 Prentice Hall 發行
  - 套件: rutebook (從 non-free)
  - 檔案: /usr/share/doc/rutebook/

參閱 'Debian 的技術支援' 227 取得更多的學習資源。

## Chapter 5

# 將 distribution 升級至 stable, testing 或 unstable

官方的升級通知發佈於 http://www.debian.org/releases/stable/releasenotes 和 http://www.debian.org/releases/testing/releasenotes (正在進行中)。 將系統升級到 stable, testing 或 unstable 必須依照以下步驟一一完成:

- 升級到 Woody (如果您的系統比 Woody 更早)
- 升級到 stable
- 升級到 testing
- 升級到 unstable

如果跳過中間的版本, Debian無法保證能升級成功。

## 5.1 從 Potato 到 Woody

以下的步驟是獨立出來介紹的,因爲 Potato 的 APT 並不支援apt\_preferences (5) 文件中的功能。

編輯/etc/apt/sources.list只留下 Woody 的來源之後,請依照下列步驟升級 APT 以及 核心程式到 Woody 的版本:

# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg

接下來請升級系統到 Woody。

```
# apt-get upgrade
# apt-get dist-upgrade
```

## 5.2 升级前的準備

您可以透過網路安裝來升級目前的發行版本到另外一個。以下説明完成的方法。 首先取得 stable 的 repositories:

# cd /etc/apt
# cp -f sources.list sources.list.old
# :>sources.list
# apt-setup noprobe

如果您想升級到 testing, 請加入 testing 的來源到該檔。如果您想升級到 unstable, 請加入 unstable 的來源到該檔。

```
# cd /etc/apt
# grep -e "^deb " sources.list >srcs
# :>sources.list
# cp -f srcs sources.list
# sed -e "s/stable/testing/" srcs >>sources.list
# sed -e "s/stable/unstable/" srcs >>sources.list
# apt-get update
# apt-get install apt apt-utils
```

請看 '體驗 Debian 套件管理' 72 中説明調效 /etc/apt/sources.list 和 /etc/apt /preferences 的藝術。

## 5.3 升级

當您按照上述的説明設定好 /etc/apt/sources.list 和 /etc/apt/preferences 之 後,您就可以進行升級的動作了。

值得注意的地方,從套件上載到unstable並移植到testing的這段時間内,會造成 Debian testing延遲安裝含有安全性修正的軟體。

參閱 'Debian 套件管理系統' 71 了解套件處理基礎,遇到問題時請參閱 'APT 升級的錯誤排除方法' 77。

#### 5.3.1 使用 dselect

如果系統裝了許多套件的-dev等套件,推薦下面使用dselect的操作方法來進行套件的精 細操作 (fine-grained package control)。

# dselect update # 在升級前要先執行這一步 # dselect select # 選擇附加的套件 當執行 dselect 時,所有你目前的套件都會被選擇, dselect 會基於 Depends, Suggests 和 Recommends 來提示你附加的套件,如果不想添加任何套件,只需輸入Q退出 dselect。

```
# dselect install
```

在安裝過程中,必須回答一些有關套件設定的問題,準備好你的筆記本花點時間處理它們。參閱'dselect'74。

使用 dselect 看看,他做得還不錯;)

#### 5.3.2 使用 apt-get

```
# apt-get update
# apt-get -t stable upgrade
# apt-get -t stable dist-upgrade
# apt-get -t testing upgrade
# apt-get -t testing dist-upgrade
# apt-get -t unstable upgrade
# apt-get -t unstable dist-upgrade
```

當您將系統升級到 Sarge 時,建議用aptitude來取代apt-get。(aptitude 接受 apt-get 許多的功能和參數,甚至包含了上述的所有功能。)

利用目前 dselect 的設定來升級的方法:

# apt-get dselect-upgrade

請參閱'套件相依性'13。

## Chapter 6

# Debian 套件管理系統

進階級套件管理工具 aptitude 是首選的 APT 前端程式。它會紀錄額外安裝的軟體並解決惱 人的相依性問題。aptitude也會移除掉不被已安裝軟體需要的套件。它內建了一個套件過濾 器,但比較難上手。

synaptic 是目前首選的以 Gtk 為 toolkit 的視窗化 APT 前端程式。它的套件過濾器就 比aptitude來的友善且簡單多了。更多的功能和支援請參閱Debian Package Tags (http: //debtags.alioth.debian.org/)。

爲了減少 Debian 檔案庫 (repository) 的網路負擔並加速您的下載速度,您可以考慮從 Debian 鏡射站台下載。

如果您的區網內需要安裝重複的套件到多台電腦上,請在使用 APT 下載套件時,考慮使用 squid 設定本地端的 HTTP proxy。必要的話,設定http\_proxy環境變數或加入 http 設定 到/etc/apt/apt.conf。

儘管 apt\_preferences (5) 的 pinning 功能十分強大,但造成的影響是難以偵錯和管理。除 非熟悉該工具才考慮採用。

在 'chroot' 115 中所敘述的方法非常適合於需要同時結合系統的穩定性和使用最新版軟體的情況。

本章是基於 Woody 之後的系統所撰寫的,有些資訊只適用於 Sarge 或更新的系統。

## 6.1 介紹

如果你沒精力閱讀完所有的開發文件,那麼先看看本章的內容,然後就開始體驗 Debian testing/unstable 的威力吧:-)

#### 6.1.1 主要的套件管理工具

dpkg - 安裝 Debian 套件的工具 apt-get - APT 安裝套件的指令 aptitude - 進階級的文字介面的 APT 前端工具 synaptic - 圖形介面的 APT 前端工具 dselect - 使用選單介面的套件管理工具 tasksel - 安裝 task

這些工具並非是為了取代對方而產生的,相反的,他們甚至能共用彼此。例如說 dselect 能搭配使用 APT 和 dpkg來安裝套件。

APT 使用 /var/lib/apt/lists/\* 來追蹤可用的套件,而 dpkg 則是使用 /var/lib/dpkg /available。如果直接用 aptitude 或類似工具來安裝套件的話,別忘了要使用 dselect 的 [U]pdate 選項,或在執行 dselect update 來更新 /var/lib/dpkg/available。

在處理套件相依性的方式上, apt-get 會自動搜尋下載相依的套件,但不會額外安裝該軟體 推薦或建議的套件。

相反地, aptitude 可以設定成是否要額外安裝"推薦"或"建議"的套件。

dselect 在套件的選擇方面提供了選單方式的操作,會列出該軟體推薦或建議的套件並個別 決定是否要安裝。請參閱'套件相依性'13。

#### 6.1.2 方便的工具

dpkg-reconfigure	_	重新設定一個已經安裝的套件 (如果它是使用 debconf 的話)
dpkg-source	_	管理套件源碼檔案
dpkg-buildpackage	_	自動重新編建套件檔案
apt-cache	_	check package archive in local cache

## 6.2 體驗 Debian 套件管理

#### 6.2.1 設定 APT

參閱'升級前的準備' 68來設定sources.list。<sup>1</sup> 請參考 'Debian 系統安裝提示' 23, '將 distribution 升級至 stable, testing 或 unstable' 67和 '應急的編輯器' 189.

#### 6.2.2 tasks 安裝

您可以安裝一個許多軟體集合的套件來規畫特定用途的 Debian 系統。而該集合就叫作 "Task"。

安裝 tasks 最簡單的方法就是在安裝系統過程中,執行 tasksel。請記得先執行

dselect update

<sup>&</sup>lt;sup>1</sup>如果您的系統以testing 或 unstable為主,您可以移除/etc/apt/sources.list 和 /etc/apt /preferences中的 stable敘述,因爲testing初使是拷貝自stable。

0

建議使用 aptitude 來安裝 tasks,而這也是它的特色之一。它能在您選擇好 tasks 並準備安裝 之前再額外刪除掉您不需要的軟體。

#### 6.2.3 aptitude

aptitude 是一套全新的套件安裝系統,類似於 dselect,不同的是針對 APT 重新設計的。 它也能當作 apt-get 另一個指令介面且完全相容 apt-get的參數喔。請參閱 aptitude(1) 和 /usr/share/doc/aptitude/README。

從您開始安裝軟體,建議使用aptitude來代替所有的安裝工具,不然會失去了aptitude所 建立的套件追蹤清單。這會使您無法移除多餘的套件。

aptitude 的功能鍵如下(大部分為小寫):

Keystroke	Action
F10	Menu
?	Help for keystroke (complete listing)
u	Update package archive information
	Mark the package to be upgraded or newly installed
-	Mark the package to be removed (keep config)
_	Mark the package to be purged (remove config)
=	Place the package on hold
U	Mark all upgradable packages to be upgraded
g	Download and install selected packages
q	Quit current screen and save changes
Х	Quit current screen and discard changes
Enter	View information about a package
С	View a package's changelog
1	Change the limit for the displayed packages
/	Search for the first match
\	Repeat the last search

如同apt-get, aptitude在安裝軟體時也會解決掉惱人的相依性問題。 aptitude 也能設定成是否要安裝軟體額外推薦或建議的軟體。 利用主畫面選單上的F10 -> Options -> Dependency handling來改變預設的安裝策略。

aptitude 的其他功能:

- aptitude 能存取所有版本的套件。
- aptitude 的動作會紀錄在/var/log/aptitude。
- aptitude 能輕鬆地追蹤本地端建立的套件並列在 "Obsolete and Locally Created Pack-ages"。
- aptitude 內建強大的收尋引擎並過濾顯示的套件。 mutt的使用者會很容易上手,因為 expression 的文法是來自於 mutt。請參閱 /usr/share/doc/aptitude/README 的 "SEARCHING, LIMITING, AND EXPRESSIONS"。
- aptitude内建 su 的功能所以一般使用者皆可以執行直到安裝或移除軟體時再取得管理者的權限。

#### 6.2.4 dselect

從 Stable 發行到現在爲止(包含Potato), dselect是主要的套件維護工具。當 Sarge 發行後, 您可以考慮使用 aptitude 來取代。

當你啓動程式時, dselect 會自動選取所有 "Required"、 "Important" 和 "Standard" 分類的套件。

雖然dselect 的使用介面有點怪,但大部分的人都已經習慣了。有四主要的指令(都是大寫的指令!):

按鍵	動作
Q	離開。確認目前所選取的並離開程式。
	(override dependencies)
R	回復 (Revert)! I did not mean it.
D	不管你 (Damn it)!我不管 dselect 怎麼想的。照我的做就是了!
U	都照建議 (sUggested) 的來做

使用 D 和 Q 可以選擇有衝突的選項。要小心地使用這些指令。

在 /etc/dpkg/dselect.cfg 中加上一行 "expert" 選項以減少干擾。

對於速度慢的機器,可以在其它速度快的機器上執行 dselect 先選好套件,然後用 apt-get install 來安裝。

#### 6.2.5 使用 APT 來維持 Debian 發行版本

請編輯/etc/apt/preferences並加入以下的説明來維持系統為 testing 版本:

```
Package: *
Pin: release a=testing
Pin-Priority: 800
Package: *
Pin: release a=stable
Pin-Priority: 600
```

要注意的是追蹤testing版本會有延誤安裝安全性修正軟體的副作用。這樣的軟體是因為上傳 到unstable並移植到testing的這段期間所造成的延誤。

更多且複雜的範例請參考 apt\_preferences(5),允許您做更多的事情,例如安裝 unstable 的套件還能把系統維持在testing。

關於限制替定軟體在特定的版本的範例可以在 examples subdirectory (http: //www.debian.org/doc/manuals/debian-reference/examples/) 找到 preferences.testing和preferences.unstable。 如果您混用不同的發行版本,例如 testing 和 stable 或 unstable 和 stable,您終究會 安裝到testing或unstable版本的核心軟體,例如libc6,而這樣的行為無法保證系統無臭 蟲存在。您必須特別小心。

另外一個例子, preferences.stable會強制降級所有的軟體到stable。

Debian 不支援將某個 套件降級到先前的發行版本。但新版的套件出問題時,重裝舊版的 套件是被允許的。您可以在本地端的 /var/cache/apt/archives/ 或遠地端的 http: //snapshot.debian.net/找到先前發行的版本。請參考'使用 dpkg 來救援' 77。

Debian 也不支援將某個 發行版本 降級到先前的版本且這樣做往往會造成很多問題。如果您願意冒險的話,也是值得嘗試看看。

#### 6.2.6 aptitude ' apt-get和apt-cache 命令

當像之前的例子一樣跟隨著 testing,您可以下列的指令來管理系統。

- aptitude update (或 apt-get update 以上動作會更新檔案庫中最新的套件列表
- aptitude upgrade (或 apt-get upgrade 或 aptitude dist-upgrade 或 apt-get dist-upgrade)
   這樣會跟隨testing版本— 他們會安裝目前testing版本上可提供升級的套件以及其相 依的軟體。<sup>2</sup>
- apt-get dselect-upgrade
   這會跟隨testing版本 上述指令升級的方法是依照dselect的處理方式來決定的。
- aptitude install package/unstable
   由 unstable distribution 安裝 package,並由 testing distribution 安裝相依的套件。
- aptitude install -t unstable package

將 unstable 的 Pin-Priority 設為 990,會由 unstable distribution 安裝 package,也從 unstable distribution 安裝相依的套件。

- apt-cache policy foo bar ... 檢查 foo bar ... 套件的狀態。
- aptitude show foo bar ... | less(或apt-cache show foo bar ... | less)
   檢查 foo bar ... 套件的資料。
- aptitude install foo=2.2.4-1
   安裝 foo 套件的特定版本 2.2.4-1。

<sup>&</sup>lt;sup>2</sup>upgrade與dist-upgrade不同的地方是對於那些升級套件的相依性問題的處理方式。請參 閱apt-get(8)來了解更多細節。 aptitude upgrade 和 aptitude dist-upgrade 是aptitude 的組合指 令。按下 e可以全螢幕顯示。

- aptitude install foo bar-安裝 foo 套件,並移除 bar 套件
- aptitude remove bar
   移除 bar 套件但不刪除設定檔。
- aptitude purge bar
   移除 bar 並刪除其設定檔。

在上面的例子中,在apt-get 中使用-u 選項會列出所有要升級的套件列表,並在動作前請示使用者。aptitude預設也是如此。下面的作法也會使 apt-get完成上述的動作:

\$ cat >> /etc/apt/apt.conf << .
// 總是列出要升級的套件 (-u)
APT::Get::Show-Upgraded "true";
.</pre>

搭配 --no-act 來模擬這些安裝,移除... 套件等動作。

## 6.3 Debian 生存指令

掌握了這些知識,就能讓你享受無窮盡的"升級"了:-)

### 6.3.1 檢查 Debian 中的 bugs 並尋求幫助

如你使用某個套件出現問題,在尋求幫助或發送錯誤報告之前請確認查看過下列網站(lynx、links和w3m都很好用):

\$ lynx http://bugs.debian.org/
\$ lynx http://bugs.debian.org/package-name # 如果你知道套件名稱
\$ lynx http://bugs.debian.org/bugnumber # 如果你知道錯誤序號

在 Google (www.google.com) 中使用關鍵字 "site:debian.org" 搜索。

如有疑問,可閱讀說明文件。設定 CDPATH 如下:

export CDPATH=.:/usr/local:/usr/share/doc

#### 並輸入

\$ cd packagename
\$ pager README.Debian # 如果存在的話
\$ mc

在 'Debian 的技術支援' 227 有更多的參考資料。

#### 6.3.2 APT 升级的错误排除方法

在升級 unstable或testing 時可能會遇到在 '升級' 68 所描述的套件相依性問題。在大多數 情況下,是因爲將要升級的套件所相依性套件不存在。這個問題可用下面的方法來解決:

# aptitude dist-upgrade

如果這也沒辦法的話,就重複使用下列之一的方法至到問題自動解決:

```
# aptitude -f upgrade # 即使遇到錯誤也繼續升級
... 或
# aptitude -f dist-upgrade # 即使遇到錯誤也繼續 dist-upgrade
```

有些升級用的 script 的確有問題,所以會持續出現狀況。通常要解決這個狀況,你最好能檢查一下這些討厭套件中的 /var/lib/dpkg/info /packagename.{post-,pre-}{install,removal} script,並執行:

# dpkg --configure -a # 設定所有部分安裝的套件

如果 script 抱怨它找不到設定檔的話,在 /etc/ 中找找對應的設定檔。如果你找到的是個有.dpkg-new 的副檔名(或是類似的東西),就把它的後級去除掉(用 mv)。

在安裝 unstable或testing 系統時也可能遇到相依性問題。可用這個方法巧妙的解決:

# aptitude -f install package # override broken dependencies

要修正這些問題,另一個可用的方法是使用 equivs 套件。請參閱 /usr/share/doc /equivs/README.Debian 和 'equivs 套件' 86。

#### 6.3.3 使用 dpkg 來救援

如果你在使用 APT 遇到死胡同,那麼可以從 Debian 鏡射站台下載套件並使用dpkg來安裝。 如果您還沒連上網路,可以鎖住 /var/cache/apt/archives/ 的快取檔案。

# dpkg -i fetchmail\_6.2.5-4\_i386.deb

如果您嘗試安裝套件卻因爲相依性問題失敗的話,請搭配--ignore-depends--force-depends或其他參數來執行dpkg。dpkg(8)有更詳盡的介紹。

#### 6.3.4 回復套件的選取狀態的資料

不論是什麼原因,如果 /var/lib/dpkg/status 亂掉了的話,則 Debian 系統會失去套件 的選取狀態的資料,這是很糟糕的事。到 /var/lib/dpkg/status-old 或 /var/backups /dpkg.status.\* 找找看舊的 /var/lib/dpkg/status 檔。

因爲 /var/backups/ 這個目錄中有很多重要的系統資料,所以把它放到分開的分割區會是個 不錯的主意。

如果連舊的 /var/lib/dpkg/status 檔也找不到了,你仍可以從 /usr/share/doc/ 目錄 來回復這些資料。

```
# ls /usr/share/doc | \
grep -v [A-Z] | \
grep -v '^texmf$' | \
grep -v '^debian$' | \
awk '{print $1 " install"}' | \
dpkg --set-selections
# dselect --expert # 重新安裝系統,如果需要的話去除一些選取
```

#### 6.3.5 在 / var 崩潰後救援系統

因爲 /var 目錄下包含了常被更動的資料,如 mail,所以比較容易會有損壞。把它放到獨立的分割區可以減少風險。如果災難發生了,你必需重建 /var 以回復 Debian 系統。

從相同或較舊版本的最簡化的 Debian 系統中取得 /var 目錄內容的架構,例如 var.tar.gz (http://people.debian.org/~osamu/pub/),將它放入受損系統的根目錄,然後

#	cd /		
#	mv var var-old	#	如果還留下有用的資料的話
#	tar xvzf var.tar.gz	#	使用 Woody 架構的檔案
#	aptitude	#	或是用 dselect

上述步驟應可使系統恢復工作。使用'回復套件的選取狀態的資料'78 中描述的技術來加快套件 選取資料的恢復。([FIXME]:這個流程需要更多的實驗來驗證。)

#### 6.3.6 把套件安装到一個無法開機的系統

用 Debian 救援磁片/CD 或是一個多重開機的 Linux 系統的其它分割區來開機進入 Linux 。請參閱 '啓動系統' 93 。在 /target 掛上這個無法開機的系統,並使用 dpkg 的 chroot 安裝模式。

# dpkg --root /target -i packagefile.deb

然後設定並解決問題。

By the way, if a broken lilo is all that prevents booting, you can boot using a standard Debian rescue disk. At boot prompt, assuming the root partition of your Linux installation is in /dev /hda12 and you want runlevel 3, enter:

boot: rescue root=/dev/hda12 3

Then you are booted into an almost fully functional system with the kernel on floppy disk. (There may be minor glitches due to lack of kernel features or modules.)

#### 6.3.7 如果 dpkg 指令壞了怎麼辦

如果 dpkg 有問題,就不能安裝任何.deb 檔了。下列的流程可幫助你來修復這個情況。(在 第一行中,你可以把 "links" 換成你喜歡的瀏覽器指令。)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/
... 下載正確的 dpkg_version_arch.deb
$ su
password: *****
# ar x dpkg_version_arch.deb
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

如果是 i386 的話,也可以用 http://packages.debian.org/dpkg。

## 6.4 Debian 神技之指令

經過這些指令的啓示,你將可由無休止的升級地獄中解放出來,到達 Debian 涅盤。:-)

#### 6.4.1 檔案中的資訊

在已安装的套件中尋找特定檔案樣板所屬的套件:

\$ dpkg {-S|--search} pattern

#### 或者搜尋 Debian 檔案庫中類似的:

\$ wget http://ftp.us.debian.org/debian/dists/sarge/Contents-i386.gz

\$ zgrep -e pattern Contents-i386.gz

或是使用特殊的套件命令:

#### 6.4.2 套件的資訊

從套件 archive 中尋找並顯示資訊。編輯 /etc/apt/sources.list 以確定 APT 指向適合的 archive(s)。如果想了解 testing/unstable 中的相對應套件與當前系統安裝的套件有何差 別,使用 apt-cache policy—會好得多。

```
# apt-get check # 更新暫存區並檢查損壞的套件
$ apt-cache search pattern # 搜尋套件的文字敘述
$ apt-cache policy package # 套件的 priority/dists 資訊
$ apt-cache show -a package # show description of package in all dists
$ apt-cache showsrc package # show description of matching source package
$ apt-cache showpkg package # 套件的除錯資訊
# dpkg --audit|-C # search for partially installed packages
$ dpkg {-s|--status} package ... # 已安裝套件的敘述
$ dpkg -1 package ... # 已安裝套件的狀態 (一行一個)
$ dpkg -L package ... # 列出套件安裝的檔案
```

apt-cache showsrc 在 Woody release 時沒有文件,但是會動:) 你也可以在下列的地方找到套件資訊(我用 mc 來瀏覽):

/var/lib/apt/lists/\*
/var/lib/dpkg/available

比較下面的檔案可以正確的了解最近幾個安裝的過程對系統造成了那些改變。

/var/lib/dpkg/status
/var/backups/dpkg.status\*

#### 6.4.3 Unattended installation with APT

For an unattended installation, 請在 /etc/apt/apt.conf 加入一行:

Dpkg::Options {"--force-confold";}

這等同於執行 aptitude -y install packagename 或 apt-get -q -y install packagename。因爲這個作法是對所有的提示都回答 "yes" 的,所以可能會造成問題,請 小心使用。參閱 apt.conf(5) 和 dpkg(1) 。

你可以在稍後用'重新設定已安裝的套件'81的方法重新設定特定的套件。

#### 6.4.4 重新設定已安裝的套件

使用下面的指令來重新設定任何已安裝的套件。

```
# dpkg-reconfigure --priority=medium package [...]
# dpkg-reconfigure --all # 重新設定所有的套件
# dpkg-reconfigure locales # 產生額外的 locales
# dpkg-reconfigure --p=low xserver-xfree86 # 重新設定 X server
```

如果你要永久設定 debconf 爲對話窗模式,請對 debconf 進行重設定。 有些程式是用做特殊設定的 scripts。<sup>3</sup>

```
- 建立 /etc/apt/sources.list
apt-setup
install-mbr - 安裝 Master Boot Record manager
          - 設定本地時區
tzconfig
gpmconfig - 設定 gpm mouse daemon
eximconfig - 設定 Exim (MTA)
            - 設定 teTeX
texconfig
apacheconfig - 設定 Apache (httpd)
cvsconfig - 設定 CVS
sndconfig
           - 設定音效系統
. . .
update-alternatives - 設定預設的指令, e.g., vim as vi
update-rc.d - System-V init script management
update-menus - Debian 選單系統
. . .
```

#### 6.4.5 移除和清除套件

移除套件但保留它的設定檔:

# aptitude remove package ...
# dpkg --remove package ...

 $<sup>^{3}</sup>$ Some \*config scripts are disappearing in the newer Sarge releases and the package configuration functionality are moved to the debconf system.

移除套件並清除所有的設定檔:

# aptitude purge package ...
# dpkg --purge package ...

#### 6.4.6 维持舊的套件

舉例來説,下面的指令可以在 dselect 和 aptitude install package 時維持 libc6 和 libc6-dev 的版本:

# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections

但這個方法擋不住 aptitude install package 。如果要在 aptitude upgrade package 或 aptitude dist-upgrade 時自動降級來維持版本的話,在 /etc/apt /preferences 中加入:

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

在這裡的 "Package:"不能用類似 "libc6\*"的用法。如果你要所有有關 glibc 的二進位套 件都維持同步的話,你需要明確地把它們都列出來。

下面的指令可以列出被維持住的套件:

dpkg --get-selections "\*"|grep -e "hold\$"

#### 6.4.7 stable/testing/unstable 混合系統

apt-show-versions 能以 distribution 來列出可用的套件版本。

```
$ apt-show-versions | fgrep /testing | wc
... 在 testing 中的套件數目
$ apt-show-versions -u
... 列出可升級的套件
$ aptitude install `apt-show-versions -u -b | fgrep /unstable`
... 將所有 unstalbe 的套件升級至各自最新的版本
```

#### 6.4.8 删除暫存的套件檔

用 APT 安裝的套件會在 /var/cache/apt/archives/ 留下暫存的套件檔。這些檔案是可以 刪除的。

- # aptitude autoclean # 只刪除無用的套件檔
- # aptitude clean # 刪除所有的暫存套件檔

#### 6.4.9 記錄/複製系統設定

要把套件選取狀態複製到本地端:

```
# dpkg --get-selections "*" >myselections # 或使用 \*
```

# debconf-get-selections > debconfsel.txt

"\*" 使 myselections 也包含標記為 "purge" 的套件。

你可將這個檔案傳到另一台電腦,並用下列的方法把它安裝起來:

```
# dselect update
```

- # debconf-set-selections < debconfsel.txt</pre>
- # dpkg --set-selections <myselections</pre>
- # apt-get -u dselect-upgrade # 或 dselect install

#### 6.4.10 把套件引入 stable 系統

將 stable 系統進行部分升級,並在些環境下以重編建的方式來使用套件,這個方法是可以 期待的。這個方法可以避免因相依性而對大量套件升級。首先,將下列來源加入 /etc/apt /sources.list:

```
deb-src http://http.us.debian.org/debian testing \
  main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
  main contrib non-free
```

因爲螢幕輸出的限制,上面 deb-src 中每一項都分成了2行,實際上在 sources.list 中它 們都應該是單行。

然後取得源碼套件並編建出本地端的套件:

\$ apt-get update # 更新源碼套件的搜尋列表
\$ apt-get source package
\$ dpkg-source -x package.dsc
\$ cd package-version
... 檢查必需的套件 (在.dsc 檔中的 Build-depends) 並也一起安裝。
你也需要 "fakeroot" 套件。
\$ dpkg-buildpackage -rfakeroot
... 或是 (no sig)
\$ dpkg-buildpackage -rfakeroot -us -uc # 如 果 需 要 的 話 , 再 使

... 然後就安裝吧 \$ su -c "dpkg -i *packagefile.deb*"

通常,你會需要安裝一些以 "-dev" 結尾的套件以滿足相依性的要求。 debsign 在 devscripts 套件中。 auto-apt 可以輕鬆解決相依性的問題。請使用 fakeroot 以避免 不必要的 root 帳號使用。

在 Woody 中,這些相依性可被簡化。例如編譯 pine 源碼套件:

# apt-get build-dep pine
# apt-get source -b pine

#### 6.4.11 本地端的套件 archive

爲了要建立相容於 APT 和 dselect 系統的本地端套件,需要建立 Packages 檔,而且套件檔 需要被放在特別的檔案目錄樹中。

可以用下列的方法來建立類似官方 Debian archive 的本地端 deb repository :

```
# aptitude install dpkg-dev
# cd /usr/local
# install -d pool # 套件實體是放在這裡
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed 's/_.*$/ priority section/' | uniq > override
# editor override # 調整 priority 和 section
# dpkg-scanpackages pool override /usr/local/ \
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF</pre>
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

不然,你也可以用一個快速但骯髒的方法來建立本地端的 deb repository:

```
# aptitude install dpkg-dev
# mkdir /usr/local/debian
```

# mv /some/where/package.deb /usr/local/debian

```
# dpkg-scanpackages /usr/local/debian /dev/null | \
gzip - > /usr/local/debian/Packages.gz
```

# echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list

把這些目錄開放給 HTTP 或 FTP 存取,並在 /etc/apt/sources.list 中設定相對應的位址,就可以遠端存取這些 archives 了。

#### 6.4.12 轉換或安裝外來的二進位套件

alien 可將 Red Hat rpm , Stampede slp , Slackware tgz 和 Solaris pkg 的二進位套件檔 案格式轉成 Debian 的 deb 套件。如果你不要使用你已安裝在系統中的套件,而想裝來自其 他 Linux distribution 的套件的話,你可以使用 alien 來將其轉成你喜愛的套件格式並安裝。 alien 也支援 LSB 套件。

#### 6.4.13 Automatically install command

auto-apt is an on-demand package installation tool.

```
$ sudo auto-apt update
... update database
$ auto-apt -x -y run
Entering auto-apt mode: /bin/bash
Exit the command to leave auto-apt mode.
$ less /usr/share/doc/med-bio/copyright # access non-existing file
... Install the package which provide this file.
... Also install dependencies
```

### 6.4.14 驗證已安裝的套件檔

debsums 以 MD5 值錯碼的方式來驗證已安裝的套件檔。有些套件並沒有可用的 MD5 值錯碼。有個可能的暫時性的修正方法提供給系統管理員:

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

per Joerg Wendland < joergland@debian.org> (untested).

#### 6.4.15 sources.list 最佳化

簡而言之,像我這樣住得離美國很遠的人,用什麼奇特的方法建出來的最佳化 sources.list 其實並沒什麼顯著的效果。還是用 apt-setup 手動選個靠近的站台吧。

apt-spy 會依 latency 和頻寬來自動產生 sources.list 檔。 netselect-apt 會建立更完整的 sources.list,但在選站台時用的是比較差的方法(比較 ping 的時間)。

- # aptitude install apt-spy
- # cd /etc/apt ; mv sources.list sources.list.org
- # apt-spy -d testing -l sources.apt

## 6.5 Debian 其它特別之處

#### 6.5.1 dpkg-divert 指令

檔案移轉是強迫 dpkg 把某個檔案安裝在一個被轉移過的位置,而非預設的位置。當 Debian 套件的 scripts 發生衝突時,可用移轉來將檔案移開。系統管理者也可以用移轉來多載套件的 設定檔,或是用在當安裝套件新版本時,包括有些沒被標記為 conffiles 而需要被 dpkg 所保留 的檔案。(參閱'保留本機設定檔'11)。

# dpkg-divert [--add] filename # 加入 "diversion"
# dpkg-divert --remove filename # 刪除 "diversion"

平時不要請使用 dpkg-divert ,除非是必要的情況下。

#### 6.5.2 equivs 套件

如果你從源碼來編譯程式,最好是能將它做成本地的 Debian 套件 (\*.deb)。最後的手段是使用 equivs。

```
Package: equivs
Priority: extra
Section: admin
Description: Circumventing Debian package dependencies
This is a dummy package which can be used to create Debian
packages, which only contain dependency information.
```

### 6.5.3 Alternative 指令

如果想用 vi 來執行 vim ,請用 update-alternatives :

Enter to keep the default[\*], or type selection number: 2

在 Debian alternatives 系統中的項目都以符號連結的方式放在 /etc/alternatives/ 。

想設定你喜歡的 X 視窗環境的話,請用 update-alternatives 來修改 /usr/bin /x-session-manager 和 /usr/bin/x-window-manager 。詳請請參閱'自定 X sessions' 133 。

/bin/sh 則就直接符號連結至 /bin/bash 或 /bin/dash 。如果要相容於舊有且 bash 化的 script 的話,用 /bin/bash 會比較好點,但用 /bin/dash 可強迫訓練你與 POSIX 相容。升級至 2.4 Linux kernel 時傾向於將其設定至 /bin/dash 。

#### 6.5.4 Runlevel

大部分的Debian套件在安裝後是被設定在 runlevel 2 到 5 時會自動執行。因此, Debian 系統 的 runlevel 2,3,4,5是沒有差別的, 而 Debian 是保留給系統管理者來設定。'自訂 RunLevel' 18說明如何自定 runlevels。這樣的 runlevels 系統是與其他有名的 GNU/Linux 發行版本完全 不同。 您或許想取消 runlevel 2 上的xdm 或 gdm 在開機後自動執行 X display 管理器。您也可 以試著修改 runlevel 3 的設定。

'Runlevels' 18 有更多 Runlevl 的資訊。

#### 6.5.5 停止 daemon 服務程式

Debian distribution 非常注重系統安全,許多 daemon 伺服程式都定位在最高安全等級,因而 在預設的安裝狀態下,系統只啓動最少的可用的服務程式。

如果你不確定你執行了什麼服務程式(像 Exim , DHCP …)的話,執行 ps aux 或檢查一下 /etc/init.d/\*和 /etc/inetd.conf的内容。同時也使用'利用 PAM 來控制登入程序'121 的方法確認一下 /etc/hosts.deny 。而 pidof 指令也是很有用的(請參閱 pidof(8))。

在最近的 Debian 版本中,預設 X11 是不允許(遠端) TCP/IP 連接的。請參閱'在TCP/IP的 環境中使用X Window' 136 。經由 SSH 來 X forwarding 也是禁止的。請參閱'連線到遠端的 X server – ssh' 137 。

## Chapter 7

# Debian下的Linux核心

Debian運用自己的方式來編譯核心及相關模組。參閱'Debian和系統核心'19。

## 7.1 核心(重新)編譯

**Debian unstable**發行版中的gcc、binutils和module-init-tools可用來編譯最新的Linux核心。這方面的官方訊息,參閱/usr/share/doc/kernel-package/README.gz文件的後半部分。

核心編譯是個很困難的議題,由于目標在不斷的變化,即使是最受人尊敬的開發者也會有不同 的見解:

**Manoj Srivastava寫到:** ---initrd 需要Debian專用cramfs修正檔。

**Herbert Xu 寫到:**不,它不需要,想使用除CRAMFS以外的文件系統只需在/etc /mkinitrd/mkinitrd.conf中設定MKIMAGE。

按照Manoj和Kent撰寫的/usr/share/doc/kernel-package/README.gz以及http://kernel-handbook.alioth.debian.org/的指導小心行事,如果要編譯最新版本的核心 請確認安裝了最新unstable版kernel-package套件。

對於單機核心編譯, initrd不是必須的。我用它是希望我新編譯的核心與相應的核心一模一樣。如果使用initrd, 請先閱讀一下mkinitrd(8)和mkinitrd.conf(5)。

#### 7.1.1 核心標頭檔

絕大多數"正常"程序不需要核心標頭檔,事實上如果直接引用它們會出錯。這些程序應該 引用那些編譯glibc所用的標頭檔,它們位於Debian系統的/usr/include/linux和/usr /include/asm目錄下。

故不要在/usr/src/linux目錄中建立指向/usr/include/linux和/usr/include/ /asm的連結,一些過時的文檔曾建議建立它們。

如果某些核心類應用程序需要特定的標頭檔,可修改Makefile(s),使其包含指向"特定標頭 檔目錄/include/linux"和"特定標頭檔目錄/include/asm"的路徑。

## 7.2 透過 proc 檔案系統調整核心

透過proc檔案系統,可在運行的狀態下進行調整Linux核心動作。

有關 /proc 檔案系統下修改核心參數的基礎知識,可參閱Linux源碼中的Documentation /sysctl/\*文件。

調整核心參數的例子,可參考/etc/init.d/networking和'無法存取某此站台的怪問題'39。

參閱 sysctl.conf(5) 了解如何透過 /etc/init.d/procps.sh 和 /proc檔案系統來設定 核心啓動組態。/etc/rcS.d/S30procps.sh 在執行時會呼叫 /etc/init.d/procps.sh script。

#### 7.2.1 開啓太多檔案

Linux核心有時會報告 "Too many open files",原因是file-max預設值(8096)太小。要解決這個問題,請用 root 身份執行下列命令:

# echo "65536" > /proc/sys/fs/file-max # for 2.2 and 2.4 kernel # echo "131072" > /proc/sys/fs/inode-max # for 2.2 kernel only

或者修改加入底下的敘述到 /etc/sysctl.conf:

file-max=65536 # for 2.2 and 2.4 kernel
inode-max=131072 # for 2.2 kernel only

#### 7.2.2 清除磁碟緩衝區時間

透過proc檔案系統來修改清除磁碟緩衝區時間。下面的操作將預設的5秒時間間隔縮短到1秒。

# echo "40 0 0 100 30000 60 0 0" > /proc/sys/vm/bdflush

這可能對文件I/O效能產生一些負面影響。但它能保證文件內容是最近1秒的,比預設的5秒更短。對日誌式檔案系統來說更是如此。

#### 7.2.3 慢速且記憶體小的機器

對某些記憶體少的舊機器來說,在proc檔案系統中打開 over-commit 功能會很有效果:

# echo 1 > /proc/sys/vm/overcommit\_memory
## 7.3 核心 2.6 和 udev

udev 是取代 /dev/ 的動態配置的檔案系統。裝置名稱可以取較短的名稱。而 2.4 核心中的 devfs 已經淘汰了。

安裝 Debian 新版的 kernel-image-2.6.NN 和 udev 就能啓動該功能了。

## **Chapter 8**

# Debian技巧

## 8.1 啓動系統

參閱 LDP BootPrompt-HOWTO (http://www.tldp.org/HOWTO/BootPrompt-HOWTO. html)獲得有關系統提示的詳細資訊。

## 8.1.1 「我忘了root密碼!」(1)

只要能存取控制台 (console) 鍵盤,不需要 root 密碼就可以啓動系統登入到 root 帳號。(假設 BIOS 或 1i1o 之類的啓動引導器在啓動時不需要密碼。)

整個過程不需要另外的開機軟碟或對 BIOS 設定進行修改。在此,「Linux」只有是一個標簽, 它代表啓動 Debian 預設值安裝的 Linux 核心。

在 lilo 啓動螢幕中,一但出現boot:(對某些系統而言,必須按下shift鍵以阻止自動啓動以 及當 lilo 使用 framebuffer 時,您必須按下"TAB"來查詢所輸入的參數),就輸入:

boot: Linux init=/bin/sh

它會指示系統啓動核心後執行 /bin/sh 而不是 init。現在你已獲得 root 特權和 root shell。 由於目前的/是以唯讀方式掛載,而其它的硬碟分區均未掛載,故你必須完成下列步驟才能獲 得一個有適當功\能的系統。

init-2.03# mount -n -o remount,rw /
init-2.03# mount -avt nonfs,noproc,nosmbfs
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow

(如果在 /etc/passwd 檔案中所有使用者的第二個網域的資料為「x」,就表明系統使用 了影子密碼,必隅婋編輯/etc/shadow。)要刪除root密碼,請編輯密碼檔案中第二個資 料網域,將它設定為空白。重啓系統不用密碼就能登入到 root。當系統啓動進入 runlevel 1 時, Debian (至少是Potato以後的版本) 需要密碼,一些較老的版本則不需要。

在 /bin/ 下裝一個小編輯器是個好習慣,因為有時 /usr 是無法存取的 (參閱'應急的編輯器' 189)。

如果安裝 sash 套服軟體,當系統無法啓動時,還可執行:

boot: Linux init=/bin/sash

當 /bin/sh 無法使用時, sash 可作為 sh 的交談式替代品,它是靜態連結,內建了許多標準 工具(在系統提示符下輸入「help」可獲得參考清單)。

## 8.1.2 「我忘了root密碼!」(2)

從急救 boot/root 磁碟 啓動系統。假設 /dev/hda3 是原始 root 分區,可用下面的方法編輯密碼檔案,與上述方法一樣容易。

# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd

與上面的方法相比,該方法的好處在於不需要知道 lilo 密碼(如果有的話)。但如果系統沒 有預先設定爲從軟式磁碟機或 CD 啓動,就需要存取 BIOS 的權限。

## 8.1.3 無法啓動系統

沒在安裝過程中製作開機軟碟?沒關係。如果 lilo 損壞了,從 Debian 安裝套件中拿出開機 軟碟,用它來啓動系統。假設你的 root 分割區在 /dev/hda12,你想進入runlevel 3,在啓動 提示符後輸入:

boot: rescue root=/dev/hda12 3

接下來,系統使用軟式磁碟機上的核心啓動,你可登入到一個幾乎擁有全部功\能的系統了。 (可能有少量特性或模組無法使用。)

如果系統已經崩潰,亦可參閱'把套件安裝到一個無法開機的系統'78。

如果想做張自訂開機軟碟,參閱急救磁碟 (rescue disk) 中的 readme.txt 文件。

#### 8.1.4 「我不想直接啓動到 X!」

玩 unstable/sid 很有趣,但在啓動進程中執行不穩定的xdm、gdm、kdm或wdm會讓你焦頭 爛額。

首先,在啓動提示符後輸入如下指令獲得 root shell:

boot: Linux vga=normal s

其中,*Linux*代表你要啓動的核心對映,「vga=normal」告訴lilo在普通VGA螢幕下執行, 「s」(或「S」)是傳給init的參數,告訴它進入單使用者模式。在提示符後輸入root密碼。

有多種方法禁用X 啓動deaemons:

- run update-rc.d -f ?dm remove; update-rc.d ?dm stop 99 1 2 3 4 5 6
- insert "exit 0" at the start of all /etc/init.d/?dm files.
- rename all /etc/rc2.d/S99?dm files to /etc/rc2.d/K99?dm.
- remove all /etc/rc2.d/S99?dm files.
- run:>/etc/X11/default-display-manager

其中,rc2.d 的檔案的編號必須與/etc/inittab中指定的runlevel一致。?dm表示您需要多 次執行這些命令來選擇xdm、gdm、kdm和wdm。

在Debian下只有第一種方法「最正確」。最後一種方法比較簡單但只適用於Debian,而且還需要使用dpkg-reconfigure重新設定 display manager。其它方法都是通用的中止daemons的方法。

你仍可在任何控制台 (console) shell中輸入startx 啓動 X。

## 8.1.5 其它用於啓動提示的技巧

使用 lilo 啓動提示,可指定系統啓動到特定的 runlevel 和組態。詳情參閱BootPrompt-HOWTO (http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html) (LDP)。

如果希望系統啓動到runlevel 4,可以1i10啓動提示後輸入:

boot: Linux 4

如果希望系統啓動到正常功\能的單使用者模式,而且你知道 root 密碼,可在 lilo啓動提示 後輸入下列任一參數。

boot: Linux S boot: Linux 1 boot: Linux -s

如果希望系統以少於實際記憶體數的記憶體啓動(也就是說機器有 64MB 記憶體,只分配 48MB 給系統使用),在1i10啓動提示符後輸入:

boot: Linux mem=48M

注意,不要指定大於實際記憶體數的記憶體,否則核心會崩潰。如果你有多於 64MB 的記憶 體,如 128MB,應在系統啓動時執行 mem=128M 或在 /etc/lilo.conf 中添加類似的指令 行,否則舊核心或使用舊 BIOS 的主板將無法使用大於64MB的記憶體。

## 8.1.6 設定 GRUB 啓動參數

GRUB 是 Hurd 計畫開發的新型啓動管理器,比 Lilo 更靈活,不過啓動參數也與之稍有不同。

grub> find /vmlinuz
grub> root (hd0,0)
grub> kernel /vmlinuz root=/dev/hda1
grub> initrd /initrd
grub> boot

## 請注意 Hurd 中的裝置名稱:

HURD/GRUB	Linux	MS-DOS/Windows
(fd0)	/dev/fd0	A:
(hd0,0)	/dev/hda1	C: (usually)
(hd0,3)	/dev/hda4	F: (usually)
(hd1,3)	/dev/hdb4	?

詳情參閱 /usr/share/doc/grub/README.Debian.gz 和 /usr/share/doc/grub-doc /html/。

## 8.2 活動記錄

## 8.2.1 記錄shell活動

比起普通的個人電腦環境,Unix環境的系統管理包括了更多細致的工作。必須掌握所有基本的 群組態方法以便進行系統故障恢復。基於X11的GUI群組態工具看上去又好又方便,但不適用 於緊急狀況。 記錄 shell 活動是個好習慣,特別是 root 使用者。

Emacs:使用M-x shell在快取區中開始記錄,使用C-x C-w將快取區中的記錄寫入檔案。 Shell:使用screen指令和「^A H」,參閱'用screen來切換 console' 112或使用script指 令:

```
$ script
Script started, file is typescript
... do whatever ...
Ctrl-D
$ col -bx <typescript >savefile
$ vi savefile
```

還可使用下面的方法:

\$ bash -i 2>&1 | tee typescript

## 8.2.2 記錄 X 操作

如果需要X套用程式的操作記錄圖,內含 xterm,可使用gimp(GUI)。它可以對每個視窗 或整個螢幕拍攝快照。還可以使用 xwd(xbase-clients)、import(imagemagick)和 scrot(scrot)。

## 8.3 拷貝及建立子目錄

這些複製和存檔指令提供了備份整個系統和檔案的基礎。the example scripts (http://www.debian.org/doc/manuals/debian-reference/examples/)的 backup 指令為一個簡單的備份 scripit 範例。

#### 8.3.1 拷貝整個子目錄的基本指令

如果想重新整理檔案群組織結構,可使用下面的方法移動檔案及檔案連結:

```
標準方法:
# cp -a /source/directory /dest/directory # requires GNU cp
# (cd /source/directory && tar cf - .) | \
(cd /dest/directory && tar xvfp -)
如果包括硬連結,則需要更嚴謹的方法:
# cd /path/to/old/directory
# find . -depth -print0 | afio -p -xv -0a /mount/point/of/new/directory
如果是遠端操作:
```

```
# (cd /source/directory && tar cf - .) | \
            ssh user@host.dom (cd /dest/directory && tar xvfp - )
如果沒有連結檔案:
# scp -pr user1@host1.dom:/source/directory \
            user2@host2.dom:/dest/directory
```

下面的有關拷貝整個子目錄的訊息由 Manoj Srivastava<srivasta@debian.org> <srivasta@debian.org>發表於debian-user@lists.debian.org。

#### 8.3.2 cp

傳統上, cp並不能真正完成這個工作,因為它既沒對符號連結進行區別對待,又不能儲存硬連結。另一件需要注意的事就是稀疏檔案(有洞的檔案)。

GNU cp克服了這缺陷,然而對於非 GNU 系統, cp仍存在問題。而且使用 cp 無法生成小巧輕便的文文件案包。

% cp -a . newdir

#### 8.3.3 tar

Tar 克服了cp在處理符號連結時出現的問題,然而, cpio可以處理特殊檔案, 傳統的tar卻不行。

對於某個有多重硬連結的檔案,tar的處理方法是只將其中一個連結拷貝到磁帶上,所以日後 你只能找回拷貝中所保留那個的連結所指的檔案;cpio會爲每個連結做一個拷貝,日後你可以 找回任意一個連結所指的檔案。

在 Potato 和 Woody 中,操作.bz2檔案的tar指令參數有所變化,所以請在script中使用--bzip2而不要簡寫為-I (Potato)或-j (Woody)。

#### 8.3.4 pax

全新的,符合POSIX (IEEE Std 1003.2-1992, pages 380-388 (section 4.48) and pages 936-940 (section E.4.48))標準的,眾望所歸的,輕便的文文件案包交互工具。pax可以讀、寫以及列出 文文件案包的成員,並能拷貝檔案目錄層次。pax的操作獨立於特定的文文件案包格式,支援 各種各樣不同的文文件案包格式。

pax工具剛剛成形,還很新。

# apt-get install pax
\$ pax -rw -p e . newdir
or
\$ find . -depth | pax -rw -p e newdir

#### 8.3.5 cpio

cpio從cpio或tar文文件案包提取/放入檔案。該文文件案包可以是硬碟上的另一個檔案,也可以是磁帶或管道。

\$ find . -depth -print0 | cpio --null --sparse -pvd new-dir

## 8.3.6 afio

afio更善於處理cpio格式的文文件案包。通常它比cpio要快,且提供了更多磁帶選項,並且 能更友好的處理有訛誤的輸入資料。它支援交談式處理多卷文文件案包。用afio製作壓縮文文 件案包比壓縮tar或cpio文文件案包更安全。在備份處理script中afio是更佳的「文文件案處 理引擎」。

\$ find . -depth -print0 | afio -px -Oa new-dir

對所有的磁帶備份我都使用afio。

## 8.4 差異備份與資料同步

要進行差異備份和資料同步可使用下列幾種方法:

- rcs:備份並進行歷史記錄,只支援文本。
- rdiff-backup:備份並進行歷史記錄。支援連結。
- pdumpfs:對檔案系統進行備份和歷史記錄。支援連結。
- rsync: 單路同步。
- unison: 雙路同步。
- CVS:多路同步伺服器備份並進行歷史記錄,只支援文本,技術成熟。參閱'Concurrent Versions System (CVS)' 195。
- arch:多路同步伺服器備份並進行歷史記錄,但內含「處於工作中的目錄」。
- subversion:多路同步伺服器備份並進行歷史記錄,專用於Apache。

有關將這些方法與文文件案包操作結合套用的討論參閱'拷貝及建立子目錄'97,有關自動進行備份的討論參閱'日程安排 (cron, at) '111。

我只講解三個較容易使用的工具。

#### 8.4.1 使用rdiff進行差異備份

rdiff-backup提供了簡單好用的方法對任何檔案內含連結進行歷史差異備份。例如要對~/目錄下的所有檔案備份到/mnt/backup:

\$ rdiff-backup --include ~/tmp/keep --exclude ~/tmp ~/ /mnt/backup

從該文文件案包中取出三天前的舊資料恢復到~/old目錄:

\$ rdiff-backup -r 3D /mnt/backup ~/old

參閱rdiff-backup(1)。

## 8.4.2 使用pdumpfs進行每日備份

pdumpfs是一種簡單的每日備份系統,與Plan9的dumpfs一樣,它每天都儲存系統快照。任何時候都可以用它來恢復到某天的系統狀態。請使用pdumpfs和cron來備份你的home目錄。

在目的地目錄中,pdumpfs以YYYY/MM/DD的方式建立系統快照。第一次執行時,它將所有來 來源檔案拷貝到快照目錄。從每二次執行起,pdumpfs只有拷貝更新的或開新檔案的檔案,對 於沒有改變的檔案用硬連結方式指向前一天的系統快照,以此來節省硬碟空間。

\$ pdumpfs src-dir dest-dir [dest-basename]

See pdumpfs(8).

## 8.4.3 使用 RCS 進行定期差異備份

Changetrack會定期對 RCS 文文件案包中基於文本格式的群組態檔案的變化進行記錄。參 閱 changetrack (1)。

# apt-get install changetrack

# vi changetrack.conf

## 8.5 系統凍結恢復

#### 8.5.1 中止一個進程

執行top看看什麼進程的活動有異常。按「P」以CPU使用率排序,「M」以記憶體使用率排 序,「k」可以中止一個進程。還有一種方法,使用 BSD 風格的ps aux | less或 System-V 風格的ps -efH | less。System V 風格的排列會顯示父進程 ID PPID,這對中止發生錯誤 的(死掉的)子進程十分有用。

知道了進程的 ID,就可使用kill中止(或發信號給)某個進程,killall的作用正如其名一樣。經常使用的信號有:

1: HUP,重啓daemon 15: TERM,普通中止 9: KILL,強令中止

## 8.5.2 Alt-SysRq

核心編譯選項「Magic SysRq key」提供系統強心針。在i386機器上按下 Alt-SysRq 群組合鍵 後,試試按下列各鍵r 0 k e i s u b,奇跡產生了:

Un'r'aw讓鍵盤從X崩潰中重生。將控制台 (console) loglevel改爲'0'以減少錯誤訊息。sa'k' (system attention key) 中止目前的虛擬控制台 (console) 的所有進程。t'e'rminate中止目前的終端設備除init外的所有進程。k'i'll中止除 init 外的所有進程。

'S'ync,'u'mount和re'b'oot幫你逃離真正的險境。

本文寫作之時,Debian 預設值安裝的核心並未將這個選項編譯進去,需要重新編譯 核心激活該功\能。詳情參閱/usr/share/doc/kernel-doc-version/Documentation /sysrq.txt.gz或/usr/src/kernel-version/Documentation/sysrq.txt.gz。

## 8.6 記住這些可愛的小指令

#### 8.6.1 Pager

less就是預設的 pager (檔案內容瀏覽器)。按「h」可獲得幫助。它比more更有用。 在shell啓動script中執行eval \$(lesspipe)或eval \$(lessfile)可以讓less活力四射。 詳情參閱/usr/share/doc/lessf/LESSOPEN。使用-R選項可輸出生癖字元and enables ANSI color escape sequences.參閱less(1)。

對於某些編碼系統 (EUC) w3m可能是更好的選擇。

#### 8.6.2 释放記憶體

free和top能讓你了解記憶體資來源的許多有用訊息。別擔心「Mem:」行中「used」的大小,看看它下面的數字(本例的數字是38792)。

\$ free •	-k # for 256MB	machine				
	total	used	free	shared	buffers	cached
Mem:	257136	230456	26680	45736	116136	75528
-/+ buf	fers/cache:	38792	218344			
Swap:	264996	0	264996			

實體記憶體的準確大小可通過grep '^Memory' /var/log/dmesg得到,本例將顯示「Memory: 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init)」。

```
Total = 262144k = 256M (1k=1024, 1M=1024k)
Free to dmesg = 256984k = Total - kernel - reserved - data - init
Free to shell = 257136k = Total - kernel - reserved - data
```

約有5MB記憶體系統不能使用,因為核心需要它。

## 8.6.3 設定時間 (BIOS)

- # date MMDDhhmmCCYY
- # hwclock --utc --systohc
- # hwclock --show

設定系統時間和硬體時間為MM/DD hh:mm, CCYY。顯示時間為本地端時間而硬體時間使用的格式為UTC。

如果硬體(BIOS)時間設定程 GMT,請修改 /etc/default/rcs 中的設定成UTC=yes。

## 8.6.4 設定時間 (NTP)

参考: Managing Accurate Date and Time HOWTO (http://www.tldp.org/HOWTO/ TimePrecision-HOWTO/index.html)。

## 擁有永久Internet連接的系統設定時間

設定系統小時鐘通過遠端伺服器自動對時:

# ntpdate server

如果你的系統擁有永久的 Internet 連接,應該將該指令加入/etc/cron.daily/。

#### 偶爾進行 Internet 連接的系統設定時間

使用 chrony 套件軟體。

## 8.6.5 如何控制 console 的特色,例如螢幕保護程式

執行下列指令來取消螢幕保護程式:

對於 Linux console:

# setterm -powersave off

啓動 kon2 (kanji) console 執行:

# kon -SaveTime 0

#### 執行 X 可執行:

```
# xset s off
或
# xset -dpms
或
# xscreensaver-command -prefs
```

參閱相關的幫助頁面來管理其餘的 console 功能。請參閱 stty(1) 來改變終端機顯示的行數。

## 8.6.6 查詢系統管理資料程式庫

Glibc提供了getent(1)查 詢 管 理 資 料 程 式 庫 的 各 類 項 目 。 例 如passwd、group、hosts、services、protocols、networks。

getent database [key ...]

## 8.6.7 禁用聲音(響鈴)

最直接的方法是拔掉 PC 喇叭;-),對於Bash shell可執行:

echo "set bell-style none">> ~/.inputrc

## 8.6.8 控制台 (console) 上的错误訊息

不想看螢幕顯示的錯誤訊息,喜好設定的方法是檢查/etc/init.d/klogd,在該script中設定KLOGD="-c 3"然後執行/etc/init.d/klogd restart。另一種方法是執行dmesg -n3。

這兒是各種錯誤層級的含義:

- 0: KERN\_EMERG, 系統無法使用
- 1: KERN\_ALERT, 必須立即執行
- 2: KERN\_CRIT, 緊急狀態
- 3: KERN\_ERR, 錯誤狀態
- 4: KERN\_WARNING, 警告狀態
- 5: KERN\_NOTICE, 正常狀態且十分重要
- 6: KERN\_INFO, 報告
- 7: KERN\_DEBUG, debug-level訊息

如果你很厭惡詳細而無用的錯誤訊息,可以試試這個小補丁shutup-abit-bp6(位 元於樣例script子目錄(http://www.debian.org/doc/manuals/debian-reference/ examples/))。

另一個該看看的地方是/etc/syslog.conf;,檢查一下是否有訊息記錄被送出到了控制台 (console) 設備。

## 8.6.9 正確設定控制台 (console) 檔案類型

在類Unix系統中, console 螢幕通常要呼叫程式庫例程,這就為使用者提供了一種獨立於終端設備的方式來更新螢幕上的文字,這是合理化的最佳選擇。參閱ncurses (3X)和terminfo (5)。

在Debian系統中,有大量預定義項目:

\$ toe | less # 所有項目
\$ toe /etc/terminfo/ | less # 使用者可再群組態的項目

#### 選擇好之後請設定 TERM 這個環境變數。

當登入到遠端 Debian 系統時,如果 xterm 的 terminfo 項目在非 Debian 的 xterm 中失效,請將終端設備檔案類型改為支援較少特性的版本如「xterm-r6」。 參閱/usr/share/doc /libncurses5/FAQ。「dumb」是terminfo的最小公分母。

## 8.6.10 恢復 console 到穩定狀態

如果執行cat some-binary-file後螢幕一片混亂(指令的返回值與你的輸入大相徑庭):

\$ reset

## 8.6.11 將 DOS 下的文字檔案轉換為Unix檔案類型

將 DOS 文字檔案(行尾=^M^J)轉換成Unix文字檔案(行尾=^J)。

# apt-get install sysutils

\$ dos2unix dosfile

## 8.6.12 利用 recode 來轉換文字檔

底下示範轉換文字檔在 DOS, Mac和Unix底下斷行的方法:

\$ recode /cl../cr <dos.txt >mac.txt
\$ recode /cr.. <mac.txt >unix.txt
\$ recode ../cl <unix.txt >dos.txt

recode依據不同字元集和介面來轉換檔案:

\$ recode charset1/surface1..charset2/surface2 \
 <input.txt >output.txt

一般常使用的字元集轉換(請參閱'locales 簡介' 151)<sup>1</sup>:

- us ASCII (7 bits)
- 11 ISO Latin-1 (ISO-8859-1, Western Europe, 8 bits)
- EUCJP EUC-JP for Japanese (Unix)
- SJIS Shift-JIS for Japanese (Microsoft)
- ISO2022JP Mail encoding for Japanese (7 bits)
- u2 UCS-2 (Universal Character Set, 2 bytes)
- u8 UTF-8 (Universal Transformation Format, 8 bits) 一般常用的介面為<sup>2</sup>:
  - /cr Carriage return as end of line (Mac text)
  - /cl Carriage return line feed as end of line (DOS text)
  - / Line feed as end of line (Unix text)
  - /d1 Human readable bytewise decimal dump
  - /x1 Human readable bytewise hexidecimal dump
  - /64 Base64 encoded text
  - /QP Quoted-Printable encoded text
- 相關的資訊請參閱info recode。

#### 也有一些特別的轉換工具:

- 字元集轉換:
  - iconv locale encoding conversions
  - konwert fancy encoding conversions
- 二進制檔案轉換:
  - uuencode 和 uudecode Unix 上有的。
  - mimencode 使用在 Mail。

#### 8.6.13 Regular-expression substitution

將所有檔案FILES...中的所有FROM\_REGEX字段取代成TO\_TEXT字段。

\$ perl -i -p -e 's/FROM\_REGEX/TO\_TEXT/g;' FILES ...

-i表示「就地編輯」,-p表示「在FILES...各檔案中循環」。如果置換很復雜,應使用參數-i.bak而非-i,這有助於發生錯誤恢復;它會將每個原始檔案儲存爲以.bak爲後綴的備份 檔案。

<sup>&</sup>lt;sup>1</sup>recode 比iconv允許更多方便的別名。

<sup>2</sup>行尾表示法:

<sup>•</sup> Carriage return means ASCII 13, ASCII 0xD, ^M, and r.

<sup>•</sup> Line feed means ASCII 10, ASCII 0xA, ^J, and n.

## 8.6.14 使用 script 來編輯檔案

下面的script將刪除 5-10 行以及 16-20 行。

```
#!/bin/bash
ed $1 <<EOF
16,20d
5,10d
w
q
EOF</pre>
```

在此,ed指令與vi指令模式下的是一樣的,從外部編輯檔案的方式使它易於script化。

## 8.6.15 提取來源檔案修改部分合併到更新包

下面的操作將根據檔案位元置,提取來源檔案的修改部分並建立統一的 diff 檔案 file.patch0 或 file.patch1:

\$ diff -u file.old file.new > file.patch0
\$ diff -u old/file new/file > file.patch1

diff 檔案(也稱修正檔案)通常用於送出程式更新。收到的修正檔案可使用下面的方法更新另 一個檔案:

\$ patch -p0 file < file.patch0
\$ patch -p1 file < file.patch1</pre>

如果有三個版本的來源代碼,使用diff3來合併效率更高:

\$ diff3 -m file.mine file.old file.yours > file

## 8.6.16 分割大檔案

\$ split -b 650m file # 將大檔案分塊成多個650MB的小檔案
\$ cat x\* >largefile # 將所有小檔案合併成一個大檔案

#### 8.6.17 從文字檔格式的表格中抽取資料

假設有一個文字檔案名爲DPL,其中存放著所有前 Debian 計畫領導人的名字和他們的上臺日期,表格格式是以空格做爲分隔的。

```
IanMurdockAugust1993BrucePerensApril1996IanJacksonJanuary1998WichertAkkermanJanuary1999BenCollinsApril2001BdaleGarbeeApril2002MartinMichlmayrMarch2003
```

AWK經常用於從這類檔案中提取資料。

```
$ awk '{ print $3 }' <DPL</pre>
                                            # month started
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL  # DPL called Ian</pre>
        Murdock August 1993
Ian
Ian
        Jackson
                  January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # When Perens started</pre>
April 1996
```

Shells 像是 Bash 也可以用來分析這類型的檔案:

\$ while read first last month year; do echo \$month done <DPL ... same output as the first Awk example

這裡, read 內建的指令使用 \$IFS (internal field separators) 裡的字元來將行拆成多個文字。如果你改變 IFS 成 ":",你就能使用 shell 來分析 /etc/passwd :

```
$ oldIFS="$IFS" # save old value
$ IFS=":"
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "osamu" ]; then
```

```
echo "$user's ID is $uid"
fi
done < /etc/passwd
osamu's ID is 1001
$ IFS="$oldIFS" # restore old value</pre>
```

(在 Awk 中,使用 FS=":"就能達成相同的功能。)

IFS也能用來拆解參數的展開、指令交換、算數運算。這些功能如果夾在單引號或雙引號之內的話,將會失效。預設的 IFS 是 <space>, <tab>, and <newline> 的組合。

請小心這個 IFS 陷阱。尤其當 shell 轉譯讀入的script時,可能會出現奇怪的問題。

## 8.6.18 精巧的管道指令辅助script

下列script做爲管道的一部分十分有用。

```
find /usr | egrep -v "/usr/var|/usr/tmp|/usr/local"
                    # find all files in /usr excluding some files
xargs -n 1 command
                    # run command for all items from stdin
xargs -n 1 echo |  # split white-space-separated items into lines
xargs echo |
                   # merge all lines into a line
                  # extract lines containing pattern
grep -e pattern|
cut -d: -f3 -|
        # extract third field separated by : (passwd file etc.)
awk '{ print $3 }' | # extract third field separated by whitespaces
awk -F'\t' '{ print $3 }' |
       # extract third field separated by tab
                    # remove backspace and expand tabs to spaces
col -bx |
expand -|
                   # expand tabs
                    # sort and remove duplicates
sort -u|
tr '\n' ' '|
                   # concatenate lines into one line
                    # remove CR
tr '\r' ''|
```

```
tr 'A-Z' 'a-z' |  # convert uppercase to lowercase
sed 's/^/# /' |  # make each line a comment
sed 's/\.ext//g' |  # remove .ext
sed -n -e 2p |  # print the second line
head -n 2 - |  # print the first 2 lines
tail -n 2 - |  # print the last 2 lines
```

## 8.6.19 查看每個檔案的輕巧 script

The following ways of looping over each file matching \*.*ext* ensures proper handling of funny file names such as ones with spaces and performs equivalent process:

• Shell loop (這個範例是PS2=""的多行風格。要打成一行的話,您必須在每一行末加入分號。)

```
for x in *.ext; do
    if test -f "$x"; then
        command "$x"
    fi
    done
```

• find and xargs combination:

find . -type f -maxdepth 1 -name '\*.ext' -print0 | \
xargs -0 -n 1 command

• find with -exec option with a command:

find . -type f -maxdepth 1 -name '\*.ext' \
 -exec command '{}' \;

• find with -exec option with a short shell script:

find . -type f -maxdepth 1 -name '\*.ext' \
 -exec sh -c "command '{}' && echo 'successful'" \;

## 8.6.20 輕巧的 Perl Script

雖然 Awk scripts 透過a2p(1)就能自動轉換成 Perl,對於只有一行的 Awk Script 到 Perl Script,直接手動修改是最方便的了。例如:

awk '(\$2=="1957") { print \$3 }' |

同等於下列任意一行:

perl -ne '@f=split; if (\$f[1] eq "1957") { print "\$f[2]\n"}' |
perl -ne 'if ((@f=split)[1] eq "1957") { print "\$f[2]\n"}' |
perl -ne '@f=split; print \$f[2] if ( \$f[1]==1957 )' |
perl -lane 'print \$F[2] if \$F[1] eq "1957"' |

其實上面各行中所有per1參數中的空格均可去掉,這得益於Perl的數字字串自動轉換功能。

perl -lane 'print\$F[2]if\$F[1]eq+1957' |

有關指令行參數的訊息可參閱perlrun(1),在http://perlgolf.sourceforge.net有更多令人著魔的Perlscript,你會感興趣的。

## 8.6.21 從網頁上取得文件或通信清單(mailing list)

下面的操作將網頁轉化為文字檔案。從網上拷貝設定檔案時十分有用。

\$ lynx -dump http://www.remote-site.com/help-info.html >textfile

links和w3m也可以這麼用,只是生成的文本樣式可能略有不同。 如果是信件清單文文件案,可使用munpack從文本獲得mime內容。

## 8.6.22 列印網頁

下面的操作將網頁內容列印成PostScript檔案或送出到印表機。

```
$ apt-get install html2ps
```

\$ html2ps URL | lpr

參閱'lpr/lpd'36。還可使用a2ps和mpage套服軟體生成PostScript檔案。

## 8.6.23 列印幫助頁面

下面的操作將幫助頁面列印成PostScript檔案或送出到印表機。

\$ man -Tps some-manpage | lpr \$ man -Tps some-manpage | mpage -2 | lpr

## 8.6.24 合併兩個Postscript或PDF檔案

可以將兩個Postscript檔案或PDF檔案合併。

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite \
   -sOutputFile=bla.ps -f fool.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite \
   -sOutputFile=bla.pdf -f fool.pdf foo2.pdf
```

## 8.6.25 指令耗時

顯示某進程的耗時

#	time	some-command	>/de	v/nul]	1					
re	eal	0m0.035s	#	time	on	wall	clock	(elapsed	real	time)
us	ser	0m0.000s	#	time	in	user	mode			
sj	/S	0m0.020s	#	time	in	kerne	el mode	e		

## 8.6.26 nice指令

使用nice(來自GNU shellutils套服軟體) 可設定指令 啓動時的nice值。renice(bsdutils)和top可以重設進程的nice值。nice值為19代表最慢的(優先等級最低的)進程;負值就「not-nice」,如-20代表非常快的(優先等級高的)進程。 只有Supervisor可以設定負nice值。

有時過度的nice值對系統弊大於利,所以使用該指令要小心。

## 8.6.27 日程安排 (cron, at)

在Linux下使用cron和at進行工作日程安排。參閱at(1)、crontab(5)、crontab(8)。

執行指令crontab -e建立或編輯crontab檔案,爲規律事務(按周期循環的事務)安排日程。 下面的一個crontab檔案樣例:

# use /bin/sh to run commands, no matter what /etc/passwd says SHELL=/bin/sh # mail any output to `paul', no matter whose crontab this is MAILTO=paul # Min Hour DayOfMonth Month DayOfWeek command (Day... are OR'ed)

```
# run at 00:05, every day
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 14:15 on the first of every month -- output mailed to paul
15 14 1 * * $HOME/bin/monthly
# run at 22:00 on weekdays(1-5), annoy Joe. % for newline, last % for cc:
0 22 * * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 * echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5 4 * * sun echo "run at 04:05 every sunday"
# run at 03:40 on the first Monday of each month
40 3 1-7 * * [ "$(date +%a)" == "Mon" ] && command -args
```

```
執行at指令爲偶然工作(只執行一次的工作)安排日程:
```

\$ echo 'command -args' | at 3:40 monday

## 8.6.28 用screen來切換 console

The screen程式容許在單一的物理終端設備或終端設備類比視窗執行多個僞終端設備,每個 僞終端設備都擁有自己的交談式shell。即便可以使用Linux僞終端設備或多個xterm視窗,研 究一下如何設定screen豐富的特性仍很有益,這些特性內含:

- 回溯歷史顯示,
- 拷貝和貼上,
- 輸出到日誌,
- 圖形入口,
- 將終端設備與整個screen階段作業分離,稍後再連接。

#### 遠端存取作業方式

If you frequently log on to a Linux machine 如果你經常從遠端終端設備登入到Linux機器或使用VT100終端設備程式,screen的detach(分離)特性將簡化你的生活。

- 1 通過拔號連接登入,執行了一個非常復雜的screen會談,開啓舊檔了好幾個視窗,有編 輯器和其它一些程式。
- 2 突然你需要離開終端設備一下,但你並不想掛斷連線中止工作。
- 3 輸入^A d離開階段作業,然後登出系統。(或是更簡單些,輸入^A DD離開階段作業並 自動登出系統)
- 4 當你回來時,需要再次登入,可輸入指令screen -r, screen會如魔法般地重新連線 上所有開啓舊檔的視窗。

## 典型的screen指令

一但開啓舊檔了screen程式,除了指令按鍵(預設值為^A)所有的鍵盤輸入都被送到目前的 視窗,所有的screen指令均按特定方式輸入:^A加一個單鍵指令[加一些參數]的。常用的指 令有:

- ^A? 顯示説明螢幕(顯示指令集)
- ^A c 建立並切換到新增視窗
- ^A n 跳到下一個視窗
- ^A p 跳到上一個視窗
- ^A 0 跳到0號視窗
- ^A w 顯示視窗清單
- ^A a 將Ctrl-A做為鍵盤輸入送出到目前的視窗
- ^A h 對目前的視窗做硬拷貝寫入到檔案
- ^A H 開始/中止將目前的視窗∨饇O錄到檔案
- ^A ^X 鎖定終端設備 (密碼保護)
- ^A d 從終端設備分離螢幕階段作業
- ^A DD 分離螢幕階段作業並結束登入

以上只是screen指令的一個很小的子集。只要是你認爲screen能幹的事,沒準它真就可以! 詳情參閱screen(1)。

#### screen階段作業中的退格鍵和Ctrl-H

在執行screen時,如果發現退格鍵和/或Ctrl-H無法正常工作,可編輯/etc/screenrc,找 到這行:

bindkey -k kb stuff "\177" 將這注釋掉(例如在句首添加「#」)。

### X下與screen等價的程式

找找xmove。參閱xmove(1)。

## 8.6.29 網路測試基礎

\$ ping yahoo.com	#	檢查Internet連接
\$ traceroute yahoo.com	#	追蹤IP封包
\$ ifconfig	#	檢查主電腦設定
\$ route -n	#	檢查路由設定
\$ dig [@dns-server.com] host		dom [{a mx any}]  less
# 檢查dns-server.com的1	105	st.dom DNS記錄
# 尋找{a/mx/any}記錄		
\$ ipchains -L -n  less 🕴	+ t	僉查包過濾(2.2 kernel)
\$ iptables -L -n  less	#	檢查包過濾(2.4 kernel)
\$ netstat -a	#	尋找系統上所有開啓舊檔的連接埠
\$ netstat -linet	#	尋找系統監聽的連接埠
\$ netstat -lntcp	#	尋找系統監聽的TCP連接埠(連接埠數字)

## 8.6.30 清理 (flush) 本地端的 Mail 佇列

想要:清理 (flush) 本地端的 Mail 佇列

# exim4 -q # 清理待讀信件 # exim4 -qf # 清理所有信件

# exim4 -qff # 清理凍結信件

-qff選項用在/etc/ppp/ip-up.d/eximscript中效果更好。在 Woody 和更舊的版本中,改用exim來取代exim4。

#### 8.6.31 删除已凍結的信件

删除本地端凍結信件並返回發生錯誤訊息:

# exim4 -Mg `mailq | grep frozen | awk '{ print \$3 }'`

在 Woody 和更舊的版本中,改用exim來取代exim4。

## 8.6.32 重新寄送mbox中的信件

如果home目錄沒有空間造成procmail不能處理信件,就需要對磁碟空間進行擴容,擴容完成後需要手動分發/var/mail/username目錄中的信件到home目錄中的分類信箱,執行:

- # /etc/init.d/exim4 stop
- # formail -s procmail </var/mail/username</pre>
- # /etc/init.d/exim4 start

在 Woody 和更舊的版本中,改用exim來取代exim4。

#### 8.6.33 清理檔案内容

要清理某些檔案如日誌檔案的内容,千萬不要使用rm刪除檔案然後再建立一個新的空檔案,因 爲在兩次操作的間隔,系統可能需要存取該檔案。下面是清理檔案内容的安全方法:

```
$ :>file-to-be-cleared
```

## 8.6.34 空檔案

下面的指令可以建立空檔案:

\$ dd if=/dev/zero of=filename bs=1k count=5 # 5KB of zero content

\$ dd if=/dev/urandom of=filename bs=1M count=7 # 7MB of random content

\$ touch filename # create OB file (if file exists, updates mtime)

例如,最實用的使用方式是從Debian啓動軟式磁碟機的shell中執行下列指令將硬碟/dev/hda的内容完全清理。

# dd if=/dev/urandom of=/dev/hda; dd if=/dev/zero of=/dev/hda

## 8.6.35 chroot

chroot程式, chroot(8), 不需要重啓系統, 就可以在單獨的系統上同時執行多個不同的GNU/Linux環境。

還可以在較快主電腦的chroot下執行某些需耗大量系統資來源記憶體的程式 如apt-get或dselect,並將較慢子機的硬碟通過NFS方式掛載到主電腦,開放讀/寫權限, 在主電腦上以chroot方式操作子機。

## 用chroot來執行不同版本的Debian

在Sarge中使用debootstrap指令很容易建構chroot Debian環境。在Sarge後續版本,你可改用cdebootstrap和適當參數來替代。例如,在一臺擁有快速Internet連接的機器的/sid-root下建立一個Sid chroot:

```
main # cd /; mkdir /sid-root
main # debootstrap sid /sid-root http://ftp.debian.org/debian/
... watch it download the whole system
main # echo "proc /sid-root/proc proc none 0 0" >> /etc/fstab
main # mount /sid-root/proc
main # mount /dev/ /sid-root/dev -o bind
main # cp /etc/hosts /sid-root/etc/hosts
main # chroot /sid-root /bin/bash
chroot # cd /dev; /sbin/MAKEDEV generic; cd -
chroot # apt-setup # set-up /etc/apt/sources.list
chroot # vi /etc/apt/sources.list # point the source to unstable
chroot # dselect # you may use aptitude, install mc and vim :-)
```

現在你就擁有了一個全功能Debian子系統,可以盡情測試而不必擔心主Debian受到不利影響。

該debootstrap套用技巧還可以實現在沒有Debian安裝盤的情況下,從另一個GNU/Linux發行版下安裝Debian。參閱http://www.debian.org/releases/stable/i386/apcs04。

## 設定chroot登入

輸入chroot /sid-root /bin/bash非常簡單,但這將沿用目前的的所有環境變量,你可 能並不希望這樣並且有時還會出問題。更好的方法是,在別的虛擬終端設備上執行另一個登入 進程,登入到chroot目錄。

在Debian系統中預設值從tty1到tty6執行Linux console,tty7執行 X Window系統,在本例中,我們將tty8設定成chroot console。按照'用chroot來執行不同版本的Debian' 115中的描述建立好chroot系統後,就可以在主系統的root shell中輸入:

#### 群組態chroot下的X

想在 chroot 下安全地執行最新版的 X 和 GNOME 嗎?完全可以!下面的例子將實現在虛擬終 端設備vt9下執行 GDM。

首先,按照'用chroot來執行不同版本的Debian' 115中描述的方法安裝好chroot系統,從主系統的root下拷貝關鍵群組態檔案到chroot系統。

```
main # cp /etc/X11/XF86Config-4 /sid-root/etc/X11/XF86Config-4
main # chroot /sid-root # or use chroot console
chroot # cd /dev; /sbin/MAKEDEV generic; cd -
chroot # apt-get install gdm gnome x-window-system
chroot # vi /etc/gdm/gdm.conf # do s/vt7/vt9/ in [servers] section
chroot # /etc/init.d/gdm start
```

在此,編輯/etc/gdm/gdm.conf來改變第一個虛擬終端裝置,使其在vt7到vt9上建立虛擬 終端設備。

現在可以很容易地能過切換Linux虛擬終端設備來實現在主系統的X環境和chroot系統的X環境 之間轉換,例如使用Ctrl-Alt-F7和Ctrl-Alt-F9。酷吧!

[FIXME] 在chroot系統下gdm的initscript中添加一條注釋和一條連結。

#### 使用chroot來執行其它發行版

很容易建立一個其它發行版的 chroot 環境。使用其它發行版的安裝程式將它們安裝到單獨的硬碟分區中。例如root分區位元於/dev/hda9:

main # cd /; mkdir /other-dist
main # mount -t ext3 /dev/hda9 /other-dist
main # chroot /other-dist /bin/bash

然後,執行'用chroot來執行不同版本的Debian' 115、'設定chroot登入' 116和'群組 態chroot下的X' 116中描述的内容。

## 使用chroot來編譯軟體

這兒有一個很特殊的chroot套服軟體pbuilder,它構造一個chroot系統並在其中編譯套服軟 體。該體系可用於檢查套服軟體編譯時關聯關係是否正確,並確保編譯生成的套服軟體中沒有 不必要的或錯誤的關聯關係。

## 8.6.36 怎樣檢查 hard links

檢查兩個檔案是否是指向同一個檔案的兩個硬連結:

#### \$ ls -li file1 file2

## 8.6.37 mount硬碟上的映象檔案

如果 file.img 檔案是硬碟內容的對映檔案,而且原始硬碟的群組態參數為 xxxx = (bytes/sector)\*(sectors/cylinder),那麼,下面的指令將其掛載到/mnt:

# mount -o loop,offset=xxxx file.img /mnt

注意絕大部分的硬碟都是 512 bytes/sector。

## 8.6.38 Samba

取得 Windows 檔案的基本方法:

可從 Linux 檢查 Samba 網路上的芳鄰:

```
# smbclient -N -L ip_address_of_your_PC | less
# nmblookup -T "*"
```

## 8.6.39 外來檔案系統的操作工具

Linux 核心支援多種外來檔案系統,想存取它們只需將其掛載到合適的檔案系統下就行了。對 某些檔案系統,還提供專門工具不需要掛載,只依靠使用者空間的程式,不需要核心提供檔案 系統支援,就能完整存取。

- mtools: for MSDOS filesystem (MS-DOS, Windows)
- cpmtools: for CP/M filesystem
- hfsutils: for HFS filesystem (native Macintosh)
- hfsplus: for HFS+ filesystem (modern Macintosh)

對於建立和檢查 MS-DOS FAT 檔案系統 dosfstools 非常有用。

## 8.7 打错字的情况

這裡示範幾個危險的動作。當使用特權帳號:root時,負面影響的程度是相當嚴重。

#### 8.7.1 rm -rf .\*

在命令列操作萬用檔名(\*號)如"rm -rf .\*"可能會造成嚴重的錯誤,因爲".\*"會參考到"." 和".."。幸運地,在Debian上的"rm"版本會很聰明地檢查檔名並取消移除"." and ".."。但這並非永遠如此。試試看底下的例子來了解萬用檔名的用法。

- "echo \*":列出現在目錄下所有非. 開頭的目錄與檔案。
- "echo .[^.]\*":列出該目錄下所有.開頭的檔案和目錄
- "echo .\*":列出上層所有的檔案和目錄以及該上層目錄。

## 8.7.2 rm /etc/passwd

由於您愚蠢地移除掉 /etc/passwd 是會造成相當嚴重的損失。Debian系統會慣例性地備份重要檔案到/var/backups/。當你回存這些檔案時,請記得更改適當的權限。

- # cp /var/backups/passwd /etc/passwd
- # chmod 644 /etc/passwd

請參閱'回復套件的選取狀態的資料'78。

## Chapter 9

# Debian 系統調整

這一章節主要是説明如何在指令介面下設定系統組態。 'Debian 系統安裝提示' 23.

如果您關心系統安全性,請閱讀 Securing Debian Manual (http://www.debian.org/ doc/manuals/securing-debian-howto/) 來建構系統的安全性,該文件也可以安裝 harden-doc 套件取得。

## 9.1 初始化系統的提示

Debian 使用 System V 的 init 系統與流程,先閱讀 'init 程式' 17 了解大概内容。

## 9.1.1 自定 init 命令稿

最簡單控管 init script 的方法是設定在/etc/default下的檔案中的系統環境變數。<sup>1</sup> 舉例 來說,/etc/default/hotplug就能控制/etc/init.d/hotplug的行為。可以自行編輯 motd, sulogin 等服務命令稿並放置在/etc/init.d/rcS 讓系統開機時即可啓動。

## 9.1.2 設定系統登入日誌(system logging)

系統日誌是由 syslogd 這支程式控制,編輯 /etc/syslog.conf 設定日誌輸出型態。 透過 colorize 這個套件可以讓日誌檔更容易辨識。 查看 syslogd(8) 和 syslog.conf(5) 可 以得到更多的訊息。

## 9.1.3 硬體最佳化(Optimizing hardware)

Debian 保留了部份硬體最佳化的設定,系統管理者可以在安裝之後進行設定。

<sup>&</sup>lt;sup>1</sup>/etc/default/下的檔案只有包含系統環境變數。當 init script 執行時會讀取對應的檔案來 改變 init script 中預設的變數。請參閱 peculiar (http://lists.debian.org/debian-devel/2003/ debian-devel-200308/msg02114.html) 來了解 Debian 選擇該目錄的原因。/etc/default/同等於 Red Hat 等系統中的 /etc/sysconfig。

- hdparm
  - 硬碟讀寫最佳化軟體,效果十分顯著。
  - 請小心使用該軟體,最好先閱讀過hdparm(8)再嚐試。
  - hdparm -tT /dev/hda : 測試 /dev/hda 的讀寫速度。
  - hdparm -q -c3 -d1 -u1 -m16 /dev/hda: 加速新型的硬碟機。(有可能照成錯誤)
- setcd
  - 光碟機讀寫最佳化軟體。.
  - setcd -x 2: 設定讀取速度為 2.倍速。
  - 閱讀 setcd(1) 可以得到更多資訊。
- setserial
  - 串列埠管理工具集。
- scsitools
  - SCSI 裝置管理工具集。
- memtest86
  - 記憶體檢測軟體。
- hwtools
  - 低階硬體管理程式的集合。
    - \* irqtune: 改變硬體的 IRQ 優先權,使快速的裝置或服務可以擁有較高的優先 權,比如加速串列埠數據機的產量達3倍等。
    - \* scanport: 掃描 ISA 裝置從 0x100 到 0x3ff 的輸入輸出位置。
    - \* inb: 從輸出輸入埠傾印資料並轉換成二進制和十六進制。
- schedutils
  - Linux 程序排程工具
  - 包括了 taskset, irqset, lsrt 以及 rt。
  - 另外還有 nice and renice(不包括),可以全盤控制程序的排程參數。

掛載檔案系統時加上 noatime 這個參數能提昇檔案讀寫的效率。 閱讀 fstab(5) 以及 mount(8)。

部份的硬體裝置可以藉由 Linux Kernel 設定 proc 檔案系統來直接設定及最佳化。 閱讀 '透過 proc 檔案系統調整核心' 90。

Debian 納入了許多設定硬體特性的軟體。大部分的軟體都是針對筆記型電腦設計。底下是 Debian 中比較有趣的軟體:

- tpconfig 觸碰版設定軟體
- apmd 進階電源管理程式
- acpi-檢視 ACPI 裝置資訊的工具
- acpid-使用 ACPI 的工具集
- lphdisk 提供 Phoenix NoteBIOS 的硬體磁區休眠功能
- sleepd-設定筆記型電腦休眠狀態的軟體
- noflushd-讓閒置的硬碟可以停止運轉以節省電源

- big-cursor 提供 X 更大的滑鼠指標
- acme 啓動筆記型電腦上的多媒體功能鍵
- tpct1 IBM ThinkPad 硬體裝置設定工具
- mwavem Mwave/ACP 數據機支援
- toshset Toshiba 筆記型電腦的硬體設定程式
- toshutils Toshiba 筆記型電腦相關工具
- sjog 支援 Sony Vaio 筆記型電腦上的 "Jog Dial" 的軟體
- spicctrl 調整 Sony Vaio 液晶螢幕亮度的軟體

ACPI 是一個全新的架構是且比 APM 要來的新且完整。

部份的套件需要特定的 Kernel 模組且已經納入最新的 Kernel Source 之中。如果硬體依然不支援,只好自行抓取修正檔並重新編譯 Kernel。

## 9.2 存取限制 (Restricting access)

#### 9.2.1 利用 PAM 來控制登入程序

PAM (Pluggable Authentication Modules) 允許管理者控管使用者登入的方式。

/etc/pam.d/*	#	PAM 控制檔案的位置
/etc/pam.d/login	#	PAM 控制登入的設定檔
/etc/security/*	#	PAM 模組參數的設定檔
/etc/securetty	#	限制 root 從 console 模式下登入的設定檔
/etc/login.defs	#	登入系統之後的環境變數設定檔

如果想允許空白密碼的使用者登入系統,可以編輯 /etc/pam.d/login 並加入以下的訊息, 此動作的風險很大,請管理者斟酌。

#auth	required	pam_unix.so	nullok
auth	required	pam_permit.sc	)

xdm, qdm...等軟體透過這樣的步驟也能允許空白密碼的使用者登入。

另一方面,使用 cracklib2 可以加強密碼安全機制,設定 /etc/pam.d/passwd 並加入以下的敘述。

```
password required pam_cracklib.so retry=3 minlen=6 difok=3
```

使用 passwd 並加入 -e 這個參數會強迫使用者在下一次登入時必須修改密碼。 閱讀 passwd(1) 以取得更多的資訊。

在 Bash shell 的環境中,可以執行 ulimit -u 1000 來設定執行程序的最大值,或者直接編輯 /etc/security/limits.conf。在該檔案中可以找到類似 core 的參數,試著編輯它即 可。 PATH 的設定可以藉由 /etc/login.defs 的内容來初始化。

PAM 的文件被包裝在 libpam-doc 之中。在這個套件包裝的文件有: Linux-PAM System Administrator's Guide, 涵蓋了如何設定 PAM 以及相關模組的説明。The Linux-PAM Application Developers' Guide 以及 The Linux-PAM Module Writers' Guide。

## 9.2.2 "為何 GNU su 不支援 wheel 群組"

這是 Richard M. Stallman 的慣用語,可以在舊式的 info su 頁面底下找到。請不要擔心,目前 Debian 中的 su 程式是使用 PAM的,所以編輯 /etc/pam.d/su 並加入 pam\_wheel.so 的敘述就可以限定那些使用者可以執行 su。底下以 Debian 中的 adm 群組當作範例;如何實現 BSD 中的 wheel 群組直接執行 su 且不需輸入密碼就能切換成 root。

# anti-RMS configuration in /etc/pam.d/su
auth required pam\_wheel.so group=adm

# Wheel members to be able to su without a password auth sufficient pam\_wheel.so trust group=adm

## 9.2.3 Purposes of standard groups

比較特殊的群組:

- 如果 pam\_wheel.so 不加上 group= 參數指定群組名稱,則 root 群組便是 su 所定義 的預設 wheel 群組。
- adm 群組可以讀取系統紀錄檔。
- cdrom 群組可以對光碟機進行存取的動作。
- floppy 群組可以讀寫控制軟碟槽。
- audio 群組可以讀寫音效裝置。
- src 群組擁有管理軟體源碼的權限,比如 /usr/src 下的檔案存取。
- staff 針對新上手的系統管理者或是系統諮詢帳號所規劃的群組,該群組的使用者擁有/usr/local 以及 /home 的讀寫權。

想要了解更多的資料,請閱讀 Securing Debian Manual (http://www.debian.org/doc/ manuals/securing-debian-howto/)的 "FAQ"。Debian Woody 可以安裝 harden-doc 套件取得該文件。base-passwd (>3.4.6) 更納入了權限的列表。

## 9.2.4 更安全的工作環境-sudo

對我而言, sudo 絕大部分是防止我做出愚笨的行為,使用 sudo 會是比直接切換成 root 更好的辦法。

安裝 sudo 之後,編輯好 /etc/sudoers(http://www.debian.org/doc/manuals/ debian-reference/examples/) 就能啓動。順便檢查 /usr/share/doc/sudo /OPTIONS中的 sudo 群組參數。

這個例子是說明"staff" 群組透過 sudo 的設定之後便能以 root 身份執行任何程式;同樣地, "src" 群組透過 sudo 設定之後便能以 root 身份執行部份程式。

使用 sudo 的好處在於一般 user 只要輸入自己的密碼就能登入並被系統監視。 且能提供部份 權限給予新上手的管理者練習。舉例來說:

\$ sudo chown -R myself:mygrp .

如果你擁有 root 密碼,那麼你能以 root 權限來執行任何命令:

```
$ su -c "shutdown -h now"
Password:
```

(我知道我應該要加強管理者帳號的 sudo 權限。但這是我自家使用的伺服器,所以不用去煩惱這個問題。)

安裝 super 這個套件能夠讓一般使用者以 root 權限執行特定指令。

#### 9.2.5 服務的存取限制

super-server, inetd, 在開機時被 /etc/rc2.d/S20inetd (for RUNLEVEL=2)呼叫並啓動, 而這個檔案是 /etc/init.d/inetd 的一個 soft link。基本上, inetd 能啓動和控制其他的服務, 減低系統負載度。

當一個服務對 inetd 送出需求時,它的通訊協定與服務可以從 /etc/protocols 以及 /etc / services 中找到,然後 inetd 便會搜尋 /etc/inetd.conf 找到適合的網際網路服務來 對應,如果是 Sun-RPC 類型的服務,便會搜尋 /etc/rpc.conf 這一個設定檔。

由於系統安全上的考量,請取消/etc/inetd.conf 中不必要或是不使用的服務。Sun-RPC 服務需要啓動 NFS 以及其他 RPC 為主的程式。

有些時候, inetd 並不會直接啓動, 而是透過 tcpd 這個 TCP/IP 服務控制程式並以 /etc / inetd.conf 中已定義的伺服器服務名稱來啓動。這個情況下, tcpd 在登入以及請求時必須使用 /etc/hosts.deny 以及 /etc/hosts.allow 進行附加的檢查, 然後在執行相對應的服務程序。

如果您在新版的 Debian 上無法進行遠端登入,請將 /etc/hosts.deny 中的 "ALL: PARA-NOID" 給註解掉。

想要得知更多的資訊,請閱讀 inetd(8), inetd.conf(5), protocols(5), services(5), tcpd(8), hosts\_access(5),以及 hosts\_options(5).

有關 Sun-RPC 更多的訊息,請參閱 rpcinfo(8), portmap(8),以及 /usr/share/doc /portmap/portmapper.txt.gz.

## 9.2.6 集中式的認證功能-LDAP

請使用 Use Lightweight Directory Access Protocol (LDAP)。 請參考:

- OpenLDAP (http://www.openldap.org/)
- OpenLDAP Admin Guide in the open1dap-guide 套件
- LDP: LDAP Linux HOWTO (http://www.tldp.org/HOWTO/LDAP-HOWTO/index. html)
- LDP: LDAP Implementation HOWTO (http://www.tldp.org/HOWTO/ LDAP-Implementation-HOWTO/index.html)
- OpenLDAP, extensive use reports (http://portal.aphroland.org/~aphro/ ldap-docs/ldap.html)

• Open LDAP with Courier IMAP and Postfix (http://alinux.washcoll.edu/docs/ plc/postfix-courier-howto.html)

## 9.3 燒錄器

ATAPI/IDE 介面的燒錄器是越來越流行了。對於系統備份或是文件容量 < 640MB 的家用 户來說可是不錯的選擇。閱讀 LDP CD-Writing-HOWTO (http://www.tldp.org/HOWTO/ CD-Writing-HOWTO.html) 能夠得到更進階的資訊喔。

## 9.3.1 简介

首先,對於光碟片來說,燒錄器將資料寫入光碟片之後,已經無法回覆光碟上的資料。如果預算不是問題,建議直接購買 SCSI 介面的燒錄器來取代 ATAPI/IDE 介面的。如果要連接 IDE 裝置,請使用 PCI 匯流排(例如,在主機板上)而不要使用 ISA 匯流排(SB16 音效卡就是使用它)

如果您的燒錄器是 IDE 介面且 Kernel 為 2.2 或是 2.4 時,則需要 IDE-SCSI 來驅動燒錄功能。 所以必須使用 SCSI 泛型驅動程式。底下有兩種方法可以達成上述功能,假設使用的是較新版 本的 Kernel(如 2001 年三月的版本)。

Kernel 2.6 能使用原本的 IDE 驅動程式來取代 IDE-SCSI 並直接使用 /dev/hdx 來存取 CD-RW 裝置。搭配 DMA 會更好。

## 9.3.2 方法一: modules + 1i1o

如果您使用的是 Debian 所包裝的 kernel-image,請在 /etc/lilo.conf 中加入以下的字 串。如果有多個選項請用空白分開:

append="hdx=ide-scsi ignore=hdx"

燒錄器的裝置代號標示法為 hdx,其中的 x 代表以下的任一種硬體裝置:

hda	for a master on the first IDE port
hdb	for a slave on the first IDE port
hdc	for a master on the second IDE port
hdd	for a slave on the second IDE port
hde hdh	for a drive on an external IDE port or ATA66/100 IDE port

完成上述設定之後,以 root 權限執行以下的命令:

# lilo
# shutdown -h now

## 9.3.3 方法二: recompile the kernel

Debian 使用 make-kpkg 來產生 kernel image。執行 make-kpkg 並加入 --append\_to\_version 的參數能夠建立多種 kernel images。請參閱 'Debian下的Linux核心' 89。

執行 make menuconfig 加入底下的 kernel 參數與設定:

- bzImage
- Exclude the IDE CD driver (not a must, but simpler to do this)
- Compile in ide-scsi and sg, or make them modules

## 9.3.4 設定步驟

以下的步驟可讓系統在開機時啓動燒錄器:

# echo ide-scsi >>/etc/modules
# echo sg >>/etc/modules
# cd /dev; ln -sf scd0 cdrom

手動啓動可以這樣做:

# modprobe ide-scsi
# modprobe sg

重新開機之後,您可以用下列方式檢查:

\$ dmesg|less
# apt-get install cdrecord
# cdrecord -scanbus

[Per Warren Dodge] 如果系統同時存在 CD-ROM 以及 CD-R/RW 的話,有時會造成 ide-scsi 和 ide-cd 的衝突。試著增加底下的設定到 /etc/modutils/aliases 並執行 update-modules 和重新開機:

pre-install ide-scsi modprobe ide-cd

這會使得 IDE 驅動程式比 ide-scsi 更早載入到核心。ide-cd 管理所有的 ATAPI CD-ROM—被指明忽略的裝置除外。剩下的裝置才由 ide-scsi 來控制。

#### 9.3.5 光碟映像檔(可開機)

如何產生一個 target-directory/ 下所有資料的光碟映像檔: cd-image.raw (可開機, Joliet TRANS.TBL-enabled format; 如果不需要光碟開機, 拿掉 -b 和 -c 參數)?先將一片開機 磁片放入軟碟機中並執行

另一個方法是產生一個開機 DOS 映像檔。假設一個開機片映像檔放在 boot.img 之下,則當軟碟開機時, CD-ROM 也會被驅動。您可以藉由 freeDOS 來達成這些動作。

光碟映像檔可以當成 loop 裝置來掛載。

# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom

#### 9.3.6 寫錄到燒錄機上(R, RW):

#### 先測試看看 (假設燒錄速度為兩倍)

# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img

如果無誤,則燒錄到 CD-R 只需執行以下指令

# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img

#### 或是燒錄到 CD-RW 的片子上

# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img

#### 對於某些 CD-RW 來說,陸續的指令會好

# nice --10 cdrecord -v blank=all speed=2 dev=0,0 disk.img

#### 接續如下

# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img

爲了預防清除 CD-RW 光碟片時出現 SCSI timeouts,則以上兩個步驟是必須的。而 nice 的值 需要特別注意。
#### 9.3.7 產生一個光碟映像檔

有些版權光碟具有防拷的保護使得在使用 dd 時無法正確拷貝 (Windows 98 便是其中之一)。 您可以試用 readcd 這個指令來拷貝, readcd 包含在 cdrecord 套件之中。利用該軟體來 複製光碟大綱以產生映像檔。如果是資料光碟,請先掛載它,並執行 df 來查詢光碟大小。 將得到的 blocks (1 block = 1024 bytes) 數目除以 2 來取得 sector 數目 (1 sector = 2048)。執行 readcd 並加入以下參數來建立映像檔並燒錄到 CD-R/RW。

# readcd dev=target,lun,scsibusno # select function 11

大部分來說,只要將以上三個參數皆設為0就可以了。通常 readcd 所檢查出來的 sectors 數目都會過多!所以改用掛載時所產生的 block, sector 數目會比較好。

dd 對於光碟複製上可能會有些小問題。第一次執行 dd 時可能會產生錯誤訊息且映像檔會失去 最末端的資料。第二次執行 dd 時如果沒指定大小,可能會產生過大的映像檔,當然,這些過 多的資料都是無用的。目前解決的辦法是在執行第二次 dd 時指定正確的大小,第一次有錯誤 時也不要退出光碟。假設由 df 查出來的大小為 46301184 blocks,則執行以下指令兩次來取 得正確的映像檔大小。(以我的經驗來說)

# dd if=/dev/cdrom of=cd.img bs=2048 count=\$((46301184/2))

#### 9.3.8 Debian 光碟映像檔

您可以在 Debian CD site (http://www.debian.org/CD/) 取得最新的 Debian 光碟訊息。

如果您有高速網路,則可以考慮網路安裝:

- 部份 floppy images (http://www.debian.org/distrib/floppyinst).
- 一個 minimal bootable CD image (http://www.debian.org/CD/netinst/).

如果您的網路不夠快,則可以考慮從 CD vendor (http://www.debian.org/CD/ vendors/)來購買 Debian 光碟。

除非您是映像檔測試者,否者請不要浪費頻寬來下載光碟映像檔(即使是使用 jigdo)。

另一個值得注意的光碟映像檔為 KNOPPIX - Live Linux Filesystem On CD (http://www.knopper.net/knoppix/index-en.html)。您可以在光碟開機之後馬上擁有 Debian 系統而不需安裝到硬碟內。

#### 9.3.9 備份您的資料到 CD-R

如何備份重要的資料到 CD-R 呢?您可以採用備份 script "backup" (http://www.debian. org/doc/manuals/debian-reference/examples/)。也請留意'拷貝及建立子目錄'97 以及'差異備份與資料同步'99。

## 9.3.10 拷貝音樂光碟

我没有测試過喔:

- # apt-get install cdrecord cdparanoia
- # cdparanoia -s -B
- # cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio \*.wav

#### 或是

- # apt-get install cdrdao #disk at once
- # cdrdao read-cd --device /dev/cdrom --paranoia-mode 3 my\_cd # read cd
- # cdrdao write --device /dev/cdrom --speed 8 my\_cd # write a new CD

cdrdao 能夠完整複製光碟(沒有音軌空隙,等...)。

#### 9.3.11 燒錄 DVD-R, DVD-RW 和 DVD+RW

燒錄 DVD 光碟有兩種方式:

- 使用 growisofs 並搭配 mkisofs。
- 請依照 /usr/share/doc/cdrecord/README.DVD.Debian 的說明重新編譯 cdrecord 並加入 dvd 支援來產生本地端建立的套件。

## 9.4 X

X Window 系統是由 Xfree86 (http://www.xfree86.org/) 所提供的。 在 Debian 上有兩個 主要版本的 X 伺服器: XFree86 版本 3.3(XF3) 以及 XFree86 版本 4.x(XF4); 兩者皆是以 X11R6 爲基礎並由 X.Org (http://www.x.org/) 所建構。

了解X的基本概念,您可以閱讀X(7),LDPXWindow-User-HOWTO(http://www.tldp. org/HOWTO/XWindow-User-HOWTO.html),以及RemoteXAppsmini-HOWTO(http:// www.tldp.org/HOWTO/mini/Remote-X-Apps.html)。您可以閱讀 xfree86-common 套件中的/usr/share/doc/xfree86-common/FAQ.gz,這是專對於Debian的設計的使 用者指南。其中Branden Robinson 有一些有趣並且有權威的對 key binding 的討論。

**'X 伺服 諾1.頁** 此軟體將本地端主機的 X window 視窗或桌面顯示到使用者的螢幕(CRT, LCD) 並接受鍵盤與滑鼠的輸入。

Y client第1頁 與 X 相容的應用軟體,可以在本地端或是遠地端的主機執行。

這些正好與一般的"server"和"client"的觀念相反。

有幾種方法可以讓 "X server" (顯示端) 接受遠端連線的 "X client" (程式端):

- xhost 方式
  - 主機列表機制(很不安全)。
  - 未加密的協定(易受到網路監聽攻擊)。
  - 如果可以, 盡量不要使用。
  - 請參閱 '連接到遠端的X server xhost' 136 以及 xhost (1x)。
- xauth 方式
  - MIT magic cookie 機制 (不算安全,但比 xhost 安全多)。
  - 未加密的協定(易受到網路監聽攻擊)。
  - 如果是本地端連線,建議採用,比ssh -X 更省資源。
  - 請參閱'在 X 裡取得 root' 138 以及 xauth(1x)。
- xdm, wdm, gdm, kdm, ... 方式
  - MIT magic cookie 機制 (不算安全,但比 xauth 安全多)。
  - 請參閱 xdm(1x) 以及 Xsecurity(7) 來建立簡單的存取權限控制。
  - 請參閱 wdm(1x), qdm(8), 和 kdm.options(5) 了解更多資訊。
  - 請閱讀 '自訂 RunLevel' 18 來了解如何在不刪除 xdm 套件時取消 xdm 並回到系統啓動的終端機畫面。
- ssh -X 方式
  - 透過安全 shell 埠號轉送機制 (很安全)。
  - 加密的傳輸協定(對於本地端使用簡直是浪費)。
  - 當遠端連線時,建議採用。
  - 請閱讀 '連線到遠端的 X server ssh' 137。

以上所有遠地端連接 X 的方式,除了 ssh 之外,都需要設定 X server 啓動 TCP/IP 連線,請 閱讀 '在TCP/IP的環境中使用 X Window' 136。

## 9.4.1 X套件

目前在 woody 上已經有許多 meta 套件能降低您安装 X 系統的難度。

- **x-window-system-core** 此 metapackage 提供最基本工作站使用的 X Window 系統。它提供了 X 函式, X 伺服器 (xserver-xfree86),以及一些字型和基本的 X clients 與工具。
- **x-window-system** 此 metapackage 提供大部分的 XFree86 計畫的 X Window System 元件, 以及一套許多有名的附屬應用程式。(包含了x-window-system-core、twm和xdm等 組件,故安裝了它就不用再安裝x-window-system-core了)

**xserver-common-v3** XFree86 3.x X 伺服器(X3) 相關的軟體和工具。

xserver-\* X3伺服器器軟體的額外套件,包含了支援對那些新的X4 伺服器(xserver-xfree86)不支援的硬體。如X4不支援某些老舊的ATI mach64卡,某些顯示卡在Woody版的X4中無法工作等等。(要獲得可用的套件,可執行 apt-cache search xserver-1less。所有這些X3伺服器均是基於xserver-common-v3的。)

大多數情況下,應該安裝x-window-system (如果要使用終端模式登入,需停用xdm,詳細 方法參閱,「我不想直接啓動到 X!」, 95。)

#### 9.4.2 X的硬體偵測

在安裝系統之前只要先安裝以下軟體就能在配置 X 時偵測週邊硬體:

- discover 硬體偵測系統。
- mdetect 滑鼠自動偵測工具。
- read-edid VESA PnP 的硬體訊息監督器。

#### 9.4.3 X伺服器

請參閱 XFree86(1x) 了解更多 X 伺服器的資訊。 從本地的終端機開啓 X 伺服器的方式:

```
$ startx -- :<display> vtXX
e.g.:
$ startx -- :1 vt8 -bpp 16
... start on vt8 connected to localhost:1 with 16 bpp mode
```

--後面接的是X伺服器的參數。

注意,在使用~/.xserverrc script 來自訂X伺服器啓動時所開啓的軟體時,請確定執行的 是真正的X伺服器。如果沒這麼做會導致X伺服器啓動過慢以及退出。例如:

#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp

#### 設定X伺服器(Version 4)

(重新)設定 XF4 伺服器,

# dpkg-reconfigure --priority=low xserver-common

# dpkg-reconfigure --priority=low xserver-xfree86

將會產生 /etc/X11/XF86Config-4 文件並使用 dexconf script 來設定 X。

#### 設定X伺服器(Version 3)

(重新)設定 XF3 伺服器,

```
# dpkg-reconfigure --priority=low xserver-common-v3
```

# dpkg-reconfigure --priority=low xserver-mach64

將會產生 /etc/X11/XF86Config 文件並使用 xf86config-3 script 來設定 X。

#### 手動設定X伺服器

在 Woody 系統上,修改/etc/X11/XF86Config-4時,請不要修改以下文字間的設定

### BEGIN DEBCONF SECTION
[snip]
### END DEBCONF SECTION

正確的作法是將自訂的內容加在定義段落之前。例如,新增顯示卡,可在設定檔開頭增加類 似下面的內容:

```
Section "Device"
 Identifier
                   "Custom Device"
 Driver
                   "ati"
 Option
                   "NoAccel"
EndSection
Section "Screen"
  Identifier "Custom Screen"
 Device
             "Custom Device"
            "Generic Monitor"
 Monitor
 DefaultDepth 24
  Subsection "Display"
   Depth
               8
               "1280x960" "1152x864" "1024x768" "800x600" "640x480"
   Modes
 EndSubsection
 Subsection "Display"
   Depth
              16
               "1280x960" "1152x864" "1024x768" "800x600" "640x480"
   Modes
 EndSubsection
  Subsection "Display"
   Depth
               24
               "1280x960" "1152x864" "1024x768" "800x600" "640x480"
   Modes
 EndSubsection
```

```
EndSection
Section "ServerLayout"
Identifier "Custom"
Screen "Custom Screen"
InputDevice "Generic Keyboard" "CoreKeyboard"
InputDevice "Configured Mouse" "CorePointer"
EndSection
```

在 Sarge (本文撰寫時爲testing)系統上,如果您想再升級時保留使用者自訂/etc/X11 /XF86Config的設定,請用 root 執行下列命令:

- # cp /etc/X11/XF86Config-4 /etc/X11/XF86Config-4.custom
- # md5sum /etc/X11/XF86Config-4 > /var/lib/xfree86/XF86Config-4.md5sum
- # dpkg-reconfigure xserver-xfree86

如果想美化字體,請依照'在 X 的 TrueType 字型' 140 的説明來修改/etc/X11 /XF86Config-4。

並檢查其餘的 X 設定。不良的螢幕設定可能會造成更多比難看字型的問題,所以請調高您的螢幕更新頻率(85 Hz 不錯,75 Hz 還可以,60 Hz 就很糟了)。

#### 9.4.4 X clients

絕大多數的 X client 軟體都能用類似下面的命令啓動:

client \$ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &

命令行中個參數的意義如下:

- -geometry WIDTHxHEIGHT+XOFF+YOFF: 視窗的初始尺寸與位置。
- -fn FONTNAME:用來顯示文字的字體。 FONTNAME 可以是:
  - a14: 普通字體
  - a24: 大型字體
  - ... (使用xlsfont查詢可用的字體。)
- -display displayname: X server 的名稱。 displayname 可以是:
  - hostname: D.S表示在名為hostname的主機的螢幕 D 上顯示的稱為 screen S;使用螢幕 D 的 X server 所間聽 TCP 埠號為 6000+D。
  - host/unix:D.S表示在host 主機的螢幕D上所顯示的稱爲 screen S; 使用螢幕 D
     的 X server 所監聽 /tmp/.X11-unix/XD (所以只能從主機 host 登入。
  - :D.S就是代表 host/unix:D.S,其中 host本地主機名稱。

預設的 X client 程式(應用端)的 displayname 可透過 DISPLAY 環境變數來設定。例如,在執行某 X client 軟體之前,執行下列命令之一就可以完成設定工作:

```
$ export DISPLAY=:0
    # 預設情況下,本地主機使用第一個 X screen
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

軟體啓動程序可以在 ~/.xinitrc 中自定。例如:

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

如 '自定 X sessions' 133所提到的, startx 不理會 Xsession 所有的常規操作。通常使用~/.xsession 僅做爲最後的辦法。 請參閱xsetroot(1x), xset(1x)以及'X 資源' 137。

## 9.4.5 X sessions

啓動 X session (X server + X client)的方式如下:

- startx: 文字介面下用來啓動 X server 與 client 的 xinit 指令集。如果 ~/.xinitrc 不存在,擇/etc/X11/Xsession 會透過執行 /etc/X11/xinit/xinitrc 時呼叫。
- xdm, gdm, kdm, or wdm: X 顯示管理程式,用來啓動 X server 與 X client 且直接在 X 下控 管帳號的登入。/etc/X11/Xsession 也會被呼叫。

請參閱'「我不想直接啓動到 X!」'95 了解從終端機啓動 X 的資訊。

## 自定 X sessions

預 設 的 啓 動 script /etc/X11/Xsession是/etc/X11/Xsession.d /50xfree86-common\_determine-startup 以 及/etc/X11/Xsession.d /99xfree86-common\_start的組合。

/etc/X11/Xsession的執行會受到/etc/X11/Xsession.options的影響,基本上,它使用exec命令執行系統中的程式,讀取的順序如下:

- 1 ~/.xsession or ~/.Xsession,如果它被定義的話。
- 2 /usr/bin/x-session-manager,如果它被定義的話。
- 3 /usr/bin/x-window-manager,如果它被定義的話。
- 4 /usr/bin/x-terminal-emulator,如果它被定義的話。

Debian alternative 系統對這些命令有詳細的說明,請參閱'Alternative 指令'86。例如:

```
# update-alternatives --config x-session-manager
... or
# update-alternatives --config x-window-manager
```

如果想定義某 X 視窗管理程式為預設,同時保留已安裝的GNOME和KDE的session管理程式,可用http://bugs.debian.org/168347 中第二個錯誤報告所附的檔案替換/etc/X11/Xsession.d/50xfree86-common\_determine-startup(我希望它能早日加到發行版中),然後按下面的方法編輯/etc/X11/Xsession.options取消Xsession管理程式:

```
# /etc/X11/Xsession.options
#
# configuration options for /etc/X11/Xsession
# See Xsession.options(5) for an explanation of the available options.
# Default enabled
allow-failsafe
allow-user-resources
allow-user-xsession
use-ssh-agent
# Default disabled (enable them by uncommenting)
do-not-use-x-session-manager
#do-not-use-x-window-manager
```

如果不想按照上述的方法修改系統,則gnome-session 以及 kdebase 所附的 X session 管理 程式就變成預設的了。移除掉他們的話,X 視窗管理程式就變成預設的視窗管理程式。(廢話, 還有更好的主意嗎?)

對於那些 /etc/X11/Xsession.options 只包含一行 allow-user-xsession 的系統 而言,任何定義了~/.xsession 或 ~/.Xsession的使用者均可以自行定義/etc/X11 /Xsession的行為。

~/.xsession 檔案中的最後一行命令, 其格式應該爲exec some-window/session-manager, 用來啓動您喜歡的X視窗/session 管理程式。

/usr/share/doc/xfree86-common/examples/xsession.gz 提供了一個相當不錯的 /usr/share/doc/xfree86-common/examples/xsession.gz 範例。

我使用它來設定每一個使用者的視窗管理員, screen 存取以及語言支援。請參閱'針對使用者啓動 X Session' 135、'在 X 裡取得 root' 138、'多語系的 X 視窗系統範例' 154。

如果您希望某些 X client 軟體可以自動啓動,請參閱'X clients' 132 的範例並寫到~/.xsession來取代~/.xinitrc。

使用者新增的 X 資源(resources)可以儲存在~/.Xresources,請參閱'X 資源'137。

使用者自訂的鍵盤與滑鼠按鍵配置檔是存放在~/.xmodmaprc。請參閱'X的按鍵和指標按鈕 對應'138。

#### 針對使用者啓動 X Session

依照 '自定 X sessions' 133 這一節所描述的,要啓動使用者自訂的 X session/window manager 可透過安裝相對應的軟體並在 ~/.xsession 的最後加入以下內容 (我偏好 blackbox/fluxbox 因為簡潔快速。):

- 預設的 X session manager
  - 參閱 'Alternative 指令' 86
  - exec /usr/bin/x-session-manager
- 預設的 X window manager
  - 参閱 'Alternative 指令' 86
  - exec /usr/bin/x-window-manager
- GNOME session manager (loaded)
  - 須安裝的套件: gnome-session
  - exec /usr/bin/gnome-session
- KDE session manager (loaded)
  - 須安裝的套件: kdebase (or kdebase3 for KDE3)
  - exec /usr/bin/kde2
- Blackbox window manager (lightweight, slick)
  - 須安裝的套件: blackbox
  - exec /usr/bin/blackbox
- Fluxbox window manager (lightweight, new blackbox)
  - 須安裝的套件: fluxbox
  - exec /usr/bin/fluxbox
- Xfce window manager (Mac OS-X, SUN CDE-like)
  - 須安裝的套件: xfce
  - exec /usr/bin/xfwm
- IceWM window manager (lightweight, GNOME alternative)
  - 須安裝的套件: icewm
  - exec /usr/bin/X11/icewm
- FVWM2 virtual window manager (lightweight, Win95-like)
  - 須安裝的套件: fvwm
  - exec /usr/bin/fvwm2
- Windowmaker window manager (somewhat NexT-like)
  - 須安裝的套件: wmaker
  - exec /usr/bin/wmaker
- Enlightenment window manager (loaded)

- 須安裝的套件: enlightenment
- exec /usr/bin/enlightenment

参閲 Window Managers for X (http://www.xwinman.org).

## 配置 KDE/GNOME

要建立完整的 KDE/GNOME 桌面環境,下列的綜合套件是很有用的:

- KDE: 安裝 kde 套件
- GNOME: 安裝 gnome 套件

使用能夠額外安裝 Recommends 的套件管理程式,如 dselect and aptitude,將會提供比 apt-get 更豐富的軟體供您選擇。

如果想從 console 登入,請確定先取消 X display managers,例如 k dm, g dm以及 w dm, 這會影響(「我不想直接啓動到 X!」'95 這一節中所提及的關聯性問題。

如果您想使用 GNOME 取代 KDE 來當視窗管理程式的話,請參閱 'Alternative 指令' 86 的方 法來設定 x-session-manager。

## 9.4.6 在TCP/IP的環境中使用X Window

因為遠端TCP/IP的連線是沒有經過加密的處理所以會有被竊聽的危險性存在,因此在最近的 幾個 Debian 版本中都會取消(disable)這個 TCP/IP socket。因此如果要遠端連線就必須使用 ssh 來建立連線(參閱 '連線到遠端的 X server - ssh' 137)。

這裡所使用的方法並不鼓勵您使用除非您所在的環境是一個在良好的防火牆後方而且只有信任的使用者會使用的安全環境中。如果是這樣請用以下的指令來設定您現在X server的設定。

# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp

移除-nolisten讓X server回復對TCP/IP的監聽。

## 9.4.7 連接到遠端的X server - xhost

xhost以主機名稱(hostnames)為權限設定是非常不安全的。如果一個TCP/IP socket的連線是 被允許的,則以下的步驟將會取消主機(host)的檢查並且允許從任何一個地方的連線。(參閱 '在TCP/IP的環境中使用X Window' 136):

\$ xhost +

你可以用以下的步驟再開啓主機的檢查:

\$ xhost -

xhost 並不會區別在遠端主機上不同的使用者。並且主機名稱(hostnames)也可以被捏造。 如果你是在一個不被信任的網路中,這個方法必須避免使用,即使是在一個高度限制的主機環 境中(例如利用PPP撥接上網的環境中)。參閱 xhost (1x).

## 9.4.8 連線到遠端的 X server - ssh

ssh 能建立一道從本地端的 X server 到遠端的應用程式伺服器的安全連線。

- •設定遠端主機的/etc/ssh/sshd\_config設定檔,將 X11Forwarding 和 AllowTcpForwarding 的值設成 yes,這免除掉命令列啓動時需要的相關參數。
- 啓動本地端主機的 X server。
- 在本地主機開啓一個 xterm 視窗。
- 執行 ssh 來建立與遠端站台的連線。

localname @ localhost \$ ssh -q -X -l loginname remotehost.domain
Password:

在遠端站台執行 X 應用程式指令。

loginname @ remotehost \$ gimp &

這個方式允許輸出遠端的 X 用户畫面,就像在本地端透過 UNIX domain socket 連接。

#### 9.4.9 X 終端機模擬器 - xterm

學習 http://dickey.his.com/xterm/xterm.faq.html 中關於 xterm 的所有東西。

#### 9.4.10 X 資源

許多舊有的 X 程式使用 X 資源資料庫來設定他們的外觀,如 xterm。~/.Xresources 用來 儲存使用者資源明細。該檔在登入時會自動合併到預設的 X 資源。系統預設的 X 資源是儲存在 /etc/X11/Xresources/\*,應用程式預設的資源檔是儲存在 /etc/X11/app-defaults /\*。使用這些設定來當作學習的起點。

這些是對妳的~/.Xresources有用的設定:

```
! Set the font to a more readable 9x15
XTerm*font: 9x15
! Display a scrollbar
XTerm*scrollBar: true
! Set the size of the buffer to 1000 lines
XTerm*saveLines: 1000
! Large kterm screen
KTerm*VT100*fontList: -*-fixed-medium-r-normal--24-*,\
-*-gothic-medium-r-normal--24-*,\
-*-mincho-medium-r-normal--24-*
```

要即使改變設定的效果,執行底下指令合併它們到資料庫:

xrdb -merge ~/.Xresources

參閱 xrdb(1x)。

#### 9.4.11 X的按键和指標按鈕對應

xmodmap 程式用來編輯和顯示鍵盤修改表和按鍵對應表,用户端程式用這些表格來轉換 X 裡的按鍵代碼事件(event keycodes)成按鍵符號(keysyms)。

```
$ xmodmap -pm
... 顯示目前的 modifier 表
$ xmodmap -pk | pager
... 顯示目前的鍵盤對應表
$ xmodmap -e "pointer = 3 2 1" # 設定滑鼠為左手慣用
$ xmodmap ~/.xmodmaprc # 讀取 ~/.xmodmaprc 設定鍵盤
```

這些通常用者的對話(session)啓動 script,~/.xsession 中執行。

如何取得按鍵代碼(keycode),在X執行 xev 並按下測試的鍵. 如何取得按鍵符號(keysym)的 意義,查詢/usr/include/X11/keysymdef.h 檔案中的巨集定義。該檔中的 #define 敘述 被命名為 XK ,這些 XK 是預留給按鍵符號(keysym)的名稱.

參閱 xmodmap(1x).

#### 9.4.12 在 X 裡取得 root

如果一個 GUI 程式在執行時需要 root 特權,使用下列的步驟就能在使用者的 X server 顯示程式的輸出。為了避免可能造成的安全性問題,**千萬不要嘗試直接從 root 帳號啓動 X server**. 利用普通使用者啓動 X server 並開啓 xterm 主控台. 然後:

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:****
# printtool &
```

當使用su 到非 root 使用者時, 確定 ~/.Xauthority 是允許群組讀取以便該非 root 使用者可以讀取.

要自動化這些指令流程,建立該使用者帳號的~/.xsession,包含底下的敘述:

```
# This makes X work when I su to the root account.
if [ -z "$XAUTHORITY" ]; then
       XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
unset XSTARTUP
# If a particular window/session manager is desired, uncomment
# the following and edit it to fit your needs.
#XSTARTUP=/usr/bin/blackbox
# This starts x-window/session-manager program
if [ -z "$XSTARTUP" ]; then
 if [ -x /usr/bin/x-session-manager ]; then
   XSTARTUP=x-session-manager
 elif [ -x /usr/bin/x-window-manager ]; then
   XSTARTUP=x-window-manager
 elif [ -x /usr/bin/x-terminal-emulator ]; then
   XSTARTUP=x-terminal-emulator
 fi
fi
# execute auto selected X window/session manager
exec $XSTARTUP
```

接著在該使用者的xterm視窗執行su (不是 su -). 現在起, GUI 程式從這個xterm啓動的能夠在該使用者的 X 視窗顯示以 root 權限執行的程式的輸出畫面. 這個技巧只要在預設的/etc /X11/Xsession被執行的情況下成立. 如果一位使用者改用~/.xinitrc或~/.xsession來 自定偏好,則上述展示的環境變數 XAUTHORITY 需要以類似的 script 來設定.

另外, sudo 可用來自動化這些指令流程:

\$ sudo xterm
... or
\$ sudo -H -s

在這裡,/root/.bashrc 必須包含:

```
if [ $SUDO_USER ]; then
    sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

即使是該使用者的家目錄是 NFS 掛載的位置, 還是能正常運作, 因為 root 不需要讀取.Xauthority 檔案.

還有幾個針對這個目的的套件: kdesu, gksu, gksudo, gnome-sudo 和 xsu. 其他方式 可用來達成類似的結果: 建立 /root/.Xauthority 的符號連結到使用者對應的位置; 使 用sux (http://fgouget.free.fr/sux/sux-readme.shtml) 的 script; 或加入 "xauth merge~USER\_RUNNING\_X/.Xauthority" 到root 的初使 script.

更多訊息請參閱 debian-devel mailing list (http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html).

## 9.4.13 在 X 的 TrueType 字型

在 XFree86-4 ,標準的 xfs 和 TrueType 字型運作良好。 如果你使用 XFree86-3 ,你必需安裝 一 third-party 字型伺服器例如 xfs-xtt。

你只需要去確認你要使用 TrueType 字型的任何應用程式是靠 libXft 或 libfreetype 連結(如果你 是使用pre-compiled .debs,你大概不需要去擔心)。

首先設定字型支援基礎配備:

• 安裝 x-ttcidfont-conf 和 defoma 套件。該套件會自動產生fonts.scale 和 fonts.dir 檔案。

# apt-get install x-ttcidfont-conf

```
• 編輯/etc/X11/XF86Config-4的Section "Files" 部份如下
Section "Files"
```

FontPath "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
FontPath "/usr/share/fonts/truetype"
FontPath "/usr/lib/X11/fonts/CID"
FontPath "/usr/lib/X11/fonts/Speedo"
FontPath "/usr/lib/X11/fonts/misc"
FontPath "/usr/lib/X11/fonts/cyrillic"
FontPath "/usr/lib/X11/fonts/100dpi:unscaled"
FontPath "/usr/lib/X11/fonts/75dpi:unscaled"
FontPath "/usr/lib/X11/fonts/Type1"

EndSection

第一行將會安裝 XFree86 來使用任何你從 Debian 套件安裝的 TrueType 字型。 自從 XFree86 支援較少 Type1 字型描繪, Type1的字型目錄就放在比較後面了。 bitmap字型的 :unscaled 方法在新的 XF4 再也不需要,但我在這裡包含它只是做確認。 為了要保存 手動修改 /etc/X11/XF86Config-4的部份,請接著 '手動設定 X 伺服器' 131 的指示。

然後安裝 DFSG 字型套件:

- 西文 TrueType 字型:
  - ttf-bitstream-vera: 一套由 Bitstream, Inc創造的高品質 TrueType 字型。<sup>2</sup>

<sup>&</sup>lt;sup>2</sup>雖然 Woody 並沒有納入,你可以從 Sarge 來安裝這些字型。

- ttf-freefont:一套開放高品質 TrueType 字型包含了 UCS 字元集。
- ttf-thryomanes:一套 unicode 的 TrueType 字型,包含了 Latin, Greek, Cyrillic 和 IPA。
- 亞洲字型:
  - tfm-arphic-bsmi00lp: Chinese Arphic "AR PL Mingti2L Big5" TrueType font TeX font metric data
  - tfm-arphic-bkai00mp: Chinese Arphic "AR PL KaitiM Big5" TrueType font TeX font metric data
  - tfm-arphic-gbsn00lp: Chinese Arphic "AR PL SungtiL GB" TrueType font TeX font metric data
  - tfm-arphic-gkai00mp: Chinese Arphic "AR PL KaitiM GB" TrueType font TeX font metric data
  - ttf-baekmuk: Korean Baekmuk 系列的 TrueType 字型
  - hbf-jfs56: Chinese Jianti Fangsong 56x56 bitmap font (GB2312) for CJK
  - hbf-cns40-b5: Chinese Fanti Song 40x40 bitmap font (Big5) for CJK
  - hbf-kanji48: Japanese Kanji 48x48 bitmap font (JIS X-0208) for CJK

自從 Free 字型有些限制開始,對 Debian 使用者來說,安裝或分享一些商業性的 TrueType 字型變成一種選擇。為了要使這程序對使用者來說更加容易,一些便利的套件被開發出來:

- ttf-commercial
- msttcorefonts (>1.1.0)<sup>3</sup>

You'll have a really good selection of TrueType fonts at the expense of contaminating your **Free** system with non-Free fonts.

所有在 Debian 的字型套件應該能無痛使用且所有的 X 程式只要使用 "core"字型系統,就應該能使用這些字型。這包含了像是 Xterm, Emacs, 和許多其他非 KDE 和非 GNOME 的應用程式。

現在,執行 xfontsel 並選擇任何在 fndry 選單的 TrueType 字型,你應該能夠在 "fmly" 選單 看到許多反灰階項目。

對 KDE2.2 和 GNOME1.4 (搭配 libgdkxft0 使得 GTK 1.2 能做出反鋸齒的效果) , 同樣地, 你 需要安裝 Xft1。Xft1 是非常過時了, 並且基本上只有 GNOME1.4 和 KDE2.2 使用。輯編 /etc /X11/XftConfig 並增加一行

dir "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"

在其它"dir"行之前。4

針對 GNOME2 和 KDE3 (Sarge 之後的版本),您需要設定 font config使得 Xft2 能找到字型。<sup>5</sup>你不應該需要對這去安裝任何額外的東西,因爲每一使用 font config 的套件已經 (間 接地) 相依於 (Depends) 它了。

首先,查閱/etc/fonts/fonts.conf。應該有類似下面這一行。如果沒有,修改/etc/fonts/local.conf 並增加

<sup>3</sup>由於 Microsoft 的網站在 8/2002 之後有更動,所以 Woody 裡的套件無法正常運作。請改用 Sarge 裡的版本。

<sup>&</sup>lt;sup>4</sup>在我機器上已經沒有任何 xft1 的東西,所以我不確定你是不是需要重新啓動 X。我似乎記得 "xftcache" 會更新 Xft1 快取,但如果某人可以對我證實會更好。

<sup>&</sup>lt;sup>5</sup>Woody 沒納入 Fontconfig。

<dir>/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType</dir>

只要在 < fontconfig> 這一行之後即可。

Fontconfig 應該能直接取得字型資訊, "fc-list"應該能列出您的新字型。另一 fontconfig 的特色是你可以僅在~/.fonts/ 中停用字型,而所有字型設定程式能夠即時接受更動。

如果你在 X 下手動安裝非 Debian 套件的新 TrueType 字型,請執行

# xset fp rehash

使得 XFree86 重新查詢目錄的内容並產生新的資料。

#### 9.4.14 在X中的網頁瀏覽器

在 Woody 發行版本中,納入了一些解析影像功能的網頁瀏覽器套件:

- mozilla Mozilla 瀏覽器 (新增的)
- galeon 以 Mozilla 為基礎, 套用 Gnome UI 的瀏覽器 (新增的)
- konqueror KDE b瀏覽器
- dillo GTK 瀏覽器
- amaya-gtk W3C 參考瀏覽器
- amaya-lesstif W3C 參考瀏覽器
- netscape-... (很多,舊式的)
- communicator-...(很多,舊式的)
- ...

mozilla 這個版本需要符合 galeon 所需要的版本。雖然他們在 UI 中不同,這兩個程式共用 Gecko HTML 的描繪引擎。

瀏覽器的外掛程式,例如 mozilla 和 galeon 可以靠著安裝 "\*.so" 到 plug-in 目錄和重新啓動瀏覽器來生效。

外掛程式資源:

- Java 外掛程式:安裝 http://java.sun.com 的 "J2SE" 二源碼。
- Flash 掛外程式:安裝 http://www.macromedia.com/software/flashplayer/ 的"Macromedia Flash Player 5"二源碼。
- freewrl: VRML 瀏覽器和 Netscape 外掛程式
- ...

## 9.5 SSH

SSH (Secure SHell) 是連接網際網路的一個安全方式。Debian 納入的 free 的 SSH 版本為 OpenSSH,由 ssh 套件取得。

#### 9.5.1 SSH 的基礎

首先,安裝 OpenSSH 伺服器和用户端程式。

# apt-get update && apt-get install ssh

如果妳需要啓動 OpenSSH 伺服器,請移除 /etc/ssh/sshd\_not\_to\_be\_run。

SSH 有兩種認證的協定:

- SSH 協定版本 1:
  - Potato 發行版本只支援該協定。
  - 可使用的認證方式:
    - \* RSAAuthentication: 以 RSA identity key 為主的使用者認證
    - \* RhostsAuthentication: 以.rhosts 的主機認證 (不安全,預設爲取消)
    - \* RhostsRSAAuthentication: .rhosts 認證與 RSA host key 的結合(預設爲取消)
    - \* ChallengeResponseAuthentication: RSA challenge-response authentication
    - \* PasswordAuthentication: 密碼為主的認證
- SSH 協定版本 2:
  - Woody 發行版本主要使用該協定。
  - 可使用的認證方式:
    - \* PubkeyAuthentication: 以 public key 爲主的使用者認證
    - \* HostbasedAuthentication: 以.rhosts或/etc/hosts.equiv 並結合 public key 的用户主機認證(預設爲取消)
    - \* ChallengeResponseAuthentication: challenge-response authentication
    - \* PasswordAuthentication: 密碼為主的認證

請特別注意這些差異,尤其是轉換到 Woody 或使用其他非 Debian 的系統。

詳情請參閱 /usr/share/doc/ssh/README.Debian.gz, ssh(1), sshd(8), ssh-agent(1),和ssh-keygen(1)。

底下是主要的設定檔案:

- /etc/ssh/ssh\_config: 預設的 SSH 用户端設定檔。 參閱 ssh(1). 注意項目:
  - Host: 限制符合匹配條件所指定的 (直到下一個 Host 保留字) 主機
  - Protocol: 指定 SSH 的協定版本。 預設為 "2,1"。
  - PreferredAuthentications: 指定 SSH2 用户端 認證方式。預設為 "hostbased, publickey, keyboard-interactive, password"。
  - PasswordAuthentication: 如果妳想採用密碼 登入,妳必須確定該欄的值不是 no。
  - ForwardX11:預設為取消。執行指令時加入"-X"參數可強制啓動。
- /etc/ssh/sshd\_config: 預設的 SSH 伺服器設定檔。 參閱 sshd(8). 注意項目:
   ListenAddress: 指定 sshd 監聽的本地端位址。 多個值是可允許的。

- AllowTcpForwarding:預設為取消。
- X11Forwarding: 預設為取消。
- \$HOME/.ssh/authorized\_keys: 該列表爲用户端以該帳號連接到主機的公開金 鑰(public key)。參閱 ssh-keygen(1)。
- \$HOME/.ssh/identity: 參閱 ssh-add(1) 和 ssh-agent(1).

底下的命令會從用户端建立 ssh 連線。

\$ ssh username@hostname.domain.ext \$ ssh -1 username@hostname.domain.ext # Force SSH version 1 \$ ssh -1 -o RSAAuthentication=no -1 username foo.host # force password on SSH1 \$ ssh -o PreferredAuthentications=password -1 username foo.host # force password on SSH2

對使用者而言, ssh 功能會比 telnet 聰明和安全 (將不會有 ^] 問題)。

#### 9.5.2 Port forwarding for SMTP/POP3 tunneling

如果妳想建立一個從本地端埠號 4025, 連接遠端伺服器埠號 25 的管線, 以及透過ssh從本地端 埠號 4110 遠端伺服器埠號110, 請在本地主機執行:

在網際網路上,這是安全建立連線到 SMTP/POP3 伺服器的方式. 設定遠端主機的 /etc/ssh /sshd\_config 檔案中 AllowTcpForwarding 項目,將其值填入 yes.

#### 9.5.3 以少許密碼連線-RSA

使用 RSAAuthentication (SSH1 protocol) 或 PubkeyAuthentication (SSH2 protocol) 可免去記 憶每一台遠端主機的登入密碼.

在遠端的系統上, 個別設定 /etc/ssh/sshd\_config 檔案中的 "RSAAuthentication yes" 或 "PubkeyAuthentication yes".

然後在本地產生認證金鑰並安裝公開金鑰到遠端主機:

```
$ ssh-keygen -t rsa  # PubkeyAuthentication: RSA key for SSH2
$ cat .ssh/id_rsa.pub | ssh userl@remote \
        "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa  # PubkeyAuthentication: DSA key for SSH2
$ cat .ssh/id_dsa.pub | ssh userl@remote \
        "cat - >>.ssh/authorized_keys"
```

執行 "ssh-keygen -p"後可更改通關密碼. 記得測試連線以驗證設定正確. 爲避免問題,使用 "ssh -v".

妳可增加額外的選項到authorized\_keys項目來限制主機並指定特定指令.詳情參閱 sshd(8).

注意, SSH2 支援 HostbasedAuthentication. 為了使它運作, 妳必須更改伺服器機器上的 /etc/ssh/sshd\_config檔案, 將 HostbasedAuthentication 設定成 yes,或者設定在用 户端機器上的 \$HOME/.ssh/config 檔案.

#### 9.5.4 與不同的 SSH 用户端程式溝通

還有其他非 Unix-like 平台的 free SSH 用户端程式.

Windows puTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/)
 (GPL)

Windows (cygwin) SSH in cygwin (http://www.cygwin.com/) (GPL)

Macintosh Classic macSSH (http://www.macssh.com/) (GPL) [注意, Mac OS X 有內建 OpenSSH; 使用 Terminal application 中的 ssh]

也請參閱 SourceForge.net, site documentation (http://www.sourceforge.net/docman/?group\_id=1),"6. CVS Instructions".

## 9.5.5 設定 ssh-agent

設定通關密語來保護妳的 SSH 認證金鑰是比較安全。如果尚未設定,執行ssh-keygen -p來 設定它。

以'以少許密碼連線 – RSA' 144描述的密碼為主的方式連線到遠端主機,將你的公開金鑰(例  $\omega: \sim/.ssh/id_rsa.pub$ )放在遠端主機的 $\sim/.ssh/authorized_keys$ 。

```
$ ssh-agent bash # 或改執行 zsh/tcsh/pdksh 程式
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/osamu/.ssh/id_rsa:
Identity added: /home/osamu/.ssh/id_rsa (/home/osamu/.ssh/id_rsa)
```

```
$ scp foo user@remote.host:foo
... no passphrase needed from here on :-)
$^D
... terminating ssh-agent session
```

針對 X server 而言,正常的 Debian 啓動的 script 會執行 ssh-agent 來當父程式。所以你只 需要執行 ssh-add 一次。

更多訊息請參閱 ssh-agent (1) 和 ssh-add (1)。

## 9.5.6 SSH 錯誤解決

如果你遇到問題,檢查設定檔的權限並加入"-v"參數來執行 ssh。

當你是 root 且有防火牆上的問題,改使用"-P"參數;這會避免使用伺服器的1-1023的埠號。

如果連線到遠端站台的ssh連線突然失效,它可能是系統管理者在測試的關係,大部份在系統維護時改變host\_key的值。確定是這個原因造成的而非他人試著駭入並竄改遠端主機的名稱,則移除掉本地端的\$HOME/.ssh/known\_hosts的host\_key項目就能重獲得連線。

## 9.6 Mail

郵件設定可分成三個種類:

- 郵件傳送代理(MTA): exim4, exim, postfix, sendmail, qmail, ssmtp, nullmailer,...
- 郵件工具:procmail,fetchmail,mailx,crm114,...
- 郵件用户代理(MUA): mutt, emacs+gnus,...

## 9.6.1 郵件傳送代理 (MTAs)

針對全功能的 MTA 來說,使用 Woody 中的 exim 和 Sarge 中的 exim4。6 參考文件:

- 針對 exim 的 exim-doc 和 exim-doc-html 套件
- 針對 exim4 的 exim4-doc-info 和 exim4-doc-html
- http://www.exim.org/

如果妳考量安全性,另一個可以接受的 MTA 是 postfix 。 sendmail和 qmail都納入 Debian 套件庫中,但不推薦。

如果妳的隨身系統,例如筆記型電腦不需要 MTA 的 relay 功能,那麼妳可考慮使用這些輕量級的套件:

• ssmtp:需要SMTP連線且支援別名,或者

<sup>6</sup>接下來的部份以 exim 爲例子。Sarge 的話則以 exim4 取代。

• nullmailer:可spool但不支援別名。 在寫本文時,我發現 exim 更適合我的工作電腦-筆記型電腦。

安裝底下不相容 exim 的套件時,妳就必須移除 exim:

# dpkg -P --force-depends exim # apt-get install nullmailer # 或 ssmtp

#### Smarthost

如果您的網路連線是消費性質的,如撥接型,而您想使用 exim4 或 exim 在您的主機上,請先 確認送出的郵件是透過 ISP 或其他供應商提供的 smarthost 機制。7 有許多好處:

- ISP 的 smarthost 通常有更可靠的連線使得 SMTP 重複寄送郵件。
- 動態 IP寄出的郵件可避免被當作撥接的 spam 列表。
- 節省寄出多個收件者的郵件的頻寬。

唯一可想到的壞處是:

- 您的 ISP SMTP 服務如果出問題時的緊急救援處理。
- 用來授課講解的實驗。
- 您的主機是工作用網路伺服器。

#### Exim 基本設定

爲了讓 exim4 或 exim 成爲您的 MTA,請設定底下的檔案:

執行 "eximconfig" 來建立和編輯 (exim)
執行 "dpkg-reconfigure exim4-config" 來建立
註解 smtp 的敘述使 exim 能獨立運行
加入改寫郵件來源位置的列表

執行 exim4 或 exim 並加入 -brw, -bf, -bF, -bV, ... 等參數可用來檢查分析器。

#### 在 Exim 下設定不存在的電郵位址的垃圾桶

在 /etc/exim/exim.conf (Woody 或更新的版本) 的 DIRECTORS 部份,在最後的地方 (localuser: director 之後) 加入清潔指揮器來處理該指揮器無法反解的位址 (per Miquel van Smoorenburg):

```
catchall:
    driver = smartuser
    new_address = webmaster@mydomain.com
```

<sup>&</sup>lt;sup>7</sup>您必須遵循這規則並套用在撥接、DSL、cable或區網等透過寬頻路由器連接網路的主機。就算您的家用主機有固定 IP,還是建議採用該方法。大部份的工作站或家用服務可歸納於這個類型。

如果需要了解每個 Virtual Domain 詳細的處理方式,請在 /etc/exim/exim.conf 加入底下的敘述(這對我有用但沒完整測試過):

然後加入 "\*" 項目到 /etc/email-addresses。

#### 在 Exim 下設定寄出郵件的位址重寫

寄出郵件的位址重寫會產生適當的 "From:" 標頭,使用 exim 能輕鬆達成該功能。只要設定 /etc/exim/exim.conf 檔案最後的部份:

```
*@host1.something.dyndns.org \
   "${if eq {${lookup{$1}lsearch{/etc/passwd}{1}{0}}} {1} \
   {$0}{$1@something.dyndns.org}}" frFs
```

這將會重寫所有符合 \*@host1.something.dyndns.org 的位址。

- 1 它搜尋 /etc/password 檔案中的使用者 (\$1) 是否為本機帳號。
- 2 如果是,將會重寫位址中(\$0) 帳號所在的部份。
- 3 如果不是,則重寫網域的部份。

## 在 Exim 下設定 SMTP 認證

部份 SMTP 服務如 yahoo.com 需要 SMTP 認證才能寄信。設定 /etc/exim/exim.conf 如下所示:

```
remote_smtp:
    driver = smtp
    authenticate_hosts = smtp.mail.yahoo.com
...
smarthost:
    driver = domainlist
    transport = remote_smtp
    route_list = "* smtp.mail.yahoo.com bydns_a"
...
plain:
    driver = plaintext
    public_name = PLAIN
    client_send = "^cmatheson3^this_is_my_password"
```

```
不要忘記最後一行的雙引號喔。
```

## 9.6.2 頻取信件1-Fetchmail

fetchmail 是一套以 daemon 模式運行的郵件頻取軟體。它會透過 ISP 來頻取 POP3 帳號的 郵件到本地端的郵件系統。設定檔案:

```
/etc/init.d/fetchmail
/etc/rc?.d/???fetchmail 預設以優先權 30 來執行 update-rc.d fetchmail
/etc/fetchmailrc 設定檔 (權限 600,擁有者爲 fetchmail)
```

Potato 版本中,敘述如何從 init.d script 以 daemon 模式啓動 fetchmail 是錯誤的(Woody 已經修正)。參閱 example scripts (http://www.debian.org/doc/manuals/ debian-reference/examples/)的/etc/init.d/fetchmail 和 /etc/fetchmailrc 檔案。

如果您的郵件標頭因爲 ISP 的郵件程式增加了 ^M 符號,請加入 "stripcr" 參數到 \$HOME /.fetchmailrc:

options fetchall no keep stripcr

#### 9.6.3 處理郵件 - Procmail

procmail 是本地端郵遞和分析程式。使用者必須先建立 \$HOME/.procmailrc 才能使用 它。範例:\_procmailrc(http://www.debian.org/doc/manuals/debian-reference/ examples/)

#### 9.6.4 使用 crm114 來處理垃圾郵件

crm114 套件提供了 /usr/share/crm114/mailfilter.crm script, 該 script 是以 CRM114 所撰寫而且題攻了一個非常有效的垃圾信分析器, 餵垃圾信給它就能訓練其處理 能力。

CRM114 是一套小和用來設計分析器的語言。搭配 grep 能發揮出更大的功能。參閱 crm(1)。

## 9.6.5 閱讀郵件 - Mutt

使用 mutt 搭配 vim 來當作郵件用户代理 (MUA)。請設定 ~/.muttrc;例如:

```
# use visual mode and "gq" to reformat quotes
set editor="vim -c 'set tw=72 et ft=mail'"
#
# header weeding taken from the manual (Sven's Draconian header weeding)
#
ignore *
```

```
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

加入底下的敘述到 /etc/mailcap 或 \$HOME/.mailcap 來顯示 HTML 郵件和 MS Word 夾 檔:

text/html; lynx -force\_html %s; needsterminal; application/msword; /usr/bin/antiword '%s'; copiousoutput; description="Microsoft Word Text"; nametemplate=%s.doc

## 9.7 Localization (l10n)

Debian 是國際化的,支援越來越多的語言和區域使用規則。下一節列出目前 Debian 支援不同 語言的差異性,下下一節則討論 localization,自訂您工作環境的步驟,允許目前選擇語言的 輸入和輸出和日期規則,數字和貨幣格式以及不同於其他區域的差異。

#### 9.7.1 localization 的基礎

有許多不同的方式來自定 localization 和 國際語言的支援。

#### Localizing 鍵盤

Debian 發行版本支援了將近兩打的鍵盤對應表。在 Woody 系統,執行底下指令重新設定鍵盤:

- dpkg-reconfigure --priority=low console-data # console
- dpkg-reconfigure --priority=low xserver-xfree86 # XF4
- dpkg-reconfigure --priority=low xserver-common-v3 # XF3

#### Localizing 資料檔案

過半數以上的 Debian 軟體支援處理非美國 ASCII 字元的資料,透過 glibc 的 locale 所提供的 LC\_CTYPE 環境變數。

- 8-bit clean:幾乎所有的程式
- 其他 Latin 字元集(例如 ISO-8859-1 or ISO-8859-2):半數以上的程式
- 雙位元組(multibyte)語言,例如中文、日文、韓文:許多新程式

#### Localizing 顯示

X 能顯示任何編碼,包括 UTF-8並支援所有字型。該列表不只支援 8-bit 字型也包括 16-bit 字型,例如中文、日文和韓文。雙位元組字元輸入法是透過'其他的 X 輸入法'157 技術達到輸入的功能。參閱'多語系的 X 視窗系統範例'154 和'支援 UTF-8 的 X 終端機'158。

日文 EUC 代碼也能在 (S)VGA graphics 主控台下透過 kon2 顯示。目前有另一個新的日文顯示 套件-jfbterm,也是利用 frame-buffer 主控台。在這些主控台環境,日文輸入法必須仰賴軟 體的提供。使用 egg 套件搭配 Emacs 和日文化的 jvim 來使用 Vim 環境。

安裝非 Unicode 字型到 X 就能在 X 下顯示任何編碼的文件。所以不用太擔心字型的編碼問題。

#### Localizing 訊息和文件

許多的文字訊息和文件已經翻譯好並納入 Debian 系統,例如錯誤訊息,標準的程式 輸出,選單和說明文件。目前説明文件支援的語言有-日文、西班牙文、芬蘭文、法 文、匈牙利文、義大利文、日文、韓文、波蘭文、葡萄牙文、中文和俄羅斯文,只要 安裝manpages-LANG(LANG 是一個以逗號分隔的兩個字母的 ISO 國家代碼列表,執行 apt-cache search manpages-|less 來取得系統提供的 Unix 說明文件) 套件就能取得。

要存取 NLS 説明文件,使用者必須設定 LC\_MESSAGE 環境變數。例如,以義大利文的説明 文件來說,LC\_MESSAGE 必須設定成 it。man program 將會搜尋 /usr/share/man/it/ 底下的義大利文的説明文件。

## 9.7.2 Locales

Debian 支援 locale 機制。 Locale 是一套允許程式提供適合的輸出和對應協定功能的技術,例如字元集、日期和時間格式、貨幣符號...等。它使用環境變數來決定適合的作法。舉例來說, 假設您安裝了美式英文和德文的區域資料到系統,許多程式的錯誤訊息可變成多語言化:

\$ LANG="en\_US" cat foo
cat: foo: No such file or directory
\$ LANG="de\_DE" cat foo
cat: foo: Datei oder Verzeichnis nicht gefunden

Glibc 的功能以函式提供給程式呼叫。參閱 locale(7)。

#### 9.7.3 locales 簡介

完整的 locale 描述包含三個部份:xx\_YY.ZZZZ。

- **xx**: ISO 639 語言代碼(英文小寫)
- YY: ISO 3166 國家代碼(英文大寫)
- ZZZZ:代碼集合,也就是字元集或編碼標示?

關於語言代碼和國家代碼,請參閱info gettext中的相關敘述。

請注意這個代碼集合部份可能被"內部標準化",為了完成跨平台的相容性,移除所有的-和轉換所有的字元爲英文小寫。典型的代碼集合為:

- UTF-8: 符合所有區域的萬國碼,大部份是 1-3 個八字元組成(new de facto standard)
- ISO-8859-1: western Europe (de facto old standard)
- **ISO-8859-2**: eastern Europe (Bosnian, Croatian, Czech, Hungarian, Polish, Romanian, Serbian, Slovak, Slovenian)
- ISO-8859-3: Maltese
- ISO-8859-5: Macedonian, Serbian
- **ISO-8859-6**: Arabic
- ISO-8859-7: Greek
- ISO-8859-8: Hebrew
- ISO-8859-9: Turkish
- ISO-8859-11: Thai (=TIS-620)
- ISO-8859-13: Latvian, Lithuanian, Maori
- ISO-8859-14: Welsh
- ISO-8859-15: western Europe with euro
- KOI8-R: Russian
- KOI8-U: Ukrainian
- CP1250: Czech, Hungarian, Polish (MS Windows origin)
- CP1251: Bulgarian, Byelorussian (MS Windows origin)
- **eucJP**: Unix style Japanese (=ujis)
- eucKR: Unix style Korean
- GB2312: Unix style Simplified Chinese (=GB, =eucCN) for zh\_CN
- **Big5**: Traditional Chinese for zh\_TW
- **sjis**: Microsoft style Japanese (Shift-JIS)

As for the meaning of basic encoding system jargons:

- **ASCII**: 7 bits (0-0x7f)
- ISO-8859-?: 8 bits (0-0xff)
- ISO-10646-1: Universal Character Set (UCS) (31 bits, 0-0x7fffffff)
- UCS-2: First 16 bit of UCS as straight 2 Octets (Unicode: 0-0xffff)
- UCS-4: UCS as straight 4 Octets (UCS: 0-0x7fffffff)
- UTF-8: UCS encoded in 1-6 Octets (mostly in 3 Octets)
- **ISO-2022**: 7 bits (0-0xff) with the escape sequence. ISO-2022-JP is the most popular encoding for the Japanese e-mail.
- EUC: 8 bits + 16 bits combination (0-0xff), Unix style
- Shift-JIS: 8 bits + 16 bits combination (0-0xff), Microsoft style.

ISO-8859-?, EUC, ISO-10646-1, UCS-2, UCS-4, 和 UTF-8 共用同一個 7-bit ASCII 字元代碼。 EUC 或 Shift-JIS 使用 high-bit 字元(0x80-0xff)來指示部份的編碼是 16-bit。UTF-8 也是使用 high-bit 字元(0x80-0xff)來指示非 7-bit 字元序列位元組而這也是大部分編碼完整的系統處理非 ASCII 字元的方式。

請注意 Unicode 實作位元組順序的差異:

- Standard UCS-2, UCS-4: big endian
- Microsoft UCS-2, UCS-4: little endian for ix86 (針對特定機器)

字元集的轉換請參閱 '利用 recode 來轉換文字檔' 104。 更多訊息請參閱 Introduction to i18n (http://www.debian.org/doc/manuals/intro-i18n/)。

## 9.7.4 啓動 locale 支援

Debian 不預先編譯所有的 locales。/usr/lib/locale 目錄下的 locales 資料(包含預設的 "C")是針對您的系統編譯出來的。如果您預期的 locale 並不存在,您有兩個作法:

- 編輯 /etc/locale.gen 加入想要的 locale, 然後以 root 權限執行 locale-gen 來編 譯它。參閱 locale-gen (8) 和 "SEE ALSO" 段落的說明文件。
- 執行 dpkg-reconfigure locales 來重新設定 locales 套件。如果尚未安裝,安裝 locales 套件會呼叫 debconf 介面讓您選擇需要的 locales 並編譯該資料庫。

## 9.7.5 啓動特定的 locale

底下的環境變數設定步驟是針對程式給予特定的 locale 資訊:

- 1 LANGUAGE:該環境變數包含了依照優先權順序的冒號分隔的 locale 名稱列表。只有 在 POSIX locale 被設定成非"C"以外的值才使用它[在 Woody 是這樣; Potato 發行版本 LANGUAGE 優先權總是高於 POSIX locale]。(GNU 擴充)
- 2 LC\_ALL:如果非空的,則該值會套用在所有的 locale 分類(POSIX.1)。通常是 ""(空的)。
- 3 LC\_\*: 如果非空的,則該值會套用在對應的分類(POSIX.1)。通常是 "C"。

LC\_\* 變數是:

- LC CTYPE:字元分類和大小寫轉換。
- LC\_COLLATE:字母字串順序比較。
- LC TIME:日期和時間格式。
- LC NUMERIC:非貨幣的數字格式。
- LC MONETARY:貨幣格式。
- LC MESSAGES: 偵錯訊息和交談回應訊息的格式。
- LC\_PAPER: 紙張大小。
- LC\_NAME: 姓名格式。
- LC\_ADDRESS:地址格式和位置資訊。
- LC\_TELEPHONE:電話號碼格式。
- LC\_MEASUREMENT:度量衡單位 (Metric or Other)。
- LC\_IDENTIFICATION:辨識 locale 資訊的資料。
- 4 LANG:如果非空的且 LC\_ALL 未定義,該值會套用在所有的 LC\_\* 分類,其值也是未 定義的。(POSIX.1) 通常是 "C"。

注意,有些程式(如 Netscape 4) 會忽略 LC\_\* 的設定。

locale 程式能顯示目前使用的 locale 設定和可獲得的 locales 資料;參閱 locale(1)。(注意:locale -a 列出系統支援的 locale;這並不表示所有的資料都編譯好!參閱'啓動 locale 支援'153。)

#### 9.7.6 ISO 8601 日期格式 locale

locale 支援國際時間標準-yyyy-mm-dd(ISO 8601 日期格式),是由 en\_DK "English in Denmark" 這個 locale 所提供,然而,這有些好笑:-)這看來只能夠用在主控台畫面下執行1s的時候。

#### 9.7.7 US (ISO-8859-1) 的範例

加入底下的敘述到~/.bash\_profile:

LC\_CTYPE=en\_US.ISO-8859-1 export LC\_CTYPE

#### 9.7.8 France with Euro sign (ISO-8859-15)

加入底下的敘述到~/.bash\_profile:

LANG=fr\_FR@euro export LANG LC\_CTYPE=fr\_FR@euro export LC\_CTYPE

按照'Localizing 鍵盤' 150 設定法式鍵盤-"AZERTY";安裝manpages-fr套件加入法文説明文件。US 的 Right-Alt 鍵在歐洲被稱為 Alt-Gr。按下該組合鍵和其他按鍵後會產生許多特別的腔調字元。例如, Alt-Gr+E 會產生 Euro 記號。

大部分的西歐語系也是同樣的設定方式。

參 閱 Debian Euro HOWTO (http://www.debian.org/doc/manuals/ debian-euro-support/) 來加入歐洲新的貨幣格式。法文的更多訊息請參閱 Utiliser et configurer Debian pour le français (http://www.debian.org/doc/manuals/fr/ debian-fr-howto/)。

#### 9.7.9 多語系的X视窗系統範例

讓我們來建立一個多語系的 X 視窗環境且同時支援日文,英文,德文和法文以及不同終端機下 支援 EUC, UTF-8 和 ISO-8859-1 的編碼。

我將會展示如何自訂 Debian 選單系統。參閱 /usr/share/doc/menu/html/index.html 的 Debian 選單系統。我也會示範建立一個 mozilla 網頁瀏覽器捷徑。<sup>8</sup>

<sup>&</sup>lt;sup>8</sup>這在個範例上,有兩個問題會影響到blackbox2003的發行版本。下指令時是在 sh -c 環境下。另外, root 不需要建立~/.menu/\*項目,改用 /etc/menu/\*取代。

- 使用 'Localization (l10n)' 150 章節所提及的方式加入日文 ja\_JP.eucJP 的 locale 和其他需要的 locale 資料。(全部)
- 安裝 Kana-to-Kanji 轉換系統和字典 (針對日文語系):
  - canna 本地端伺服器 ("free-beer" license), 或
  - freewnn-jserver 加入網路功能的伺服器 (Public Domain)
- 安裝日文輸入法系統(針對日文語系):
  - kinput2-canna-X下或
  - kinput2-canna-wnn-X下和
  - egg 在主控台下就能搭配 Emacsen 直接輸入日文(額外的)
- 安裝相容的終端機(針對所有語系):
  - xterm-X(支援 ISO-8859-1 and UTF-8),
  - kterm-X(支援日文 EUC),和
  - mlterm-X(支援多語系)。
- 加入所有日文字體套件。(針對所有語系)
- 依照 '自定 X sessions' 133 章節提及的方式,建立 ~/.xsession 並設定使用者專屬的 X 環境(針對所有語系):

```
#!/bin/sh
# This makes X work when I su to root.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
# Set specific environment through debian menu system.
# Reset locale
unset LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
unset LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
unset LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
# set locale default in X
LANG=C
# export locale
export LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
export LC PAPER LC NAME LC ADDRESS LC TELEPHONE LC MEASUREMENT
export LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
###
# activate input method for Japanese with kinput2
kinput2 &
XMODIFIERS="@im=kinput2"
export XMODIFIERS
# How about blackbox window manager (lightweight)
exec blackbox
```

#exec xfwm
#exec wmaker

- 針對 Linux 主控台設定 ~/.bash\_profile 的 locale 資訊 (針對所有語系)。
- 移除~/.bashrc 已有的 locale 設定 (針對所有語系)。

```
• 建立新的檔案到/etc/menu/(針對所有語系)。
   - /etc/menu/xterm-local:(加入新項目到選單)<sup>9</sup>
            ?package(xterm):\
             needs=x11
             section=XShells\
             longtitle="XTerm: terminal emulator (en_US.ISO-8859-1)"\
             title="XTerm (en_US.ISO-8859-1)"\
             command="sh -c 'LC ALL=en US.ISO-8859-1 xterm'"
             ?package(xterm):\
             needs=x11
             section=XShells\
             longtitle="XTerm: terminal emulator (de_DE.ISO-8859-1)"\
             title="XTerm (de_DE.ISO-8859-1)"\
             command="sh -c 'LC_ALL=de_DE.ISO-8859-1 xterm -T xterm-de'"
             ?package(xterm):\
             needs=x11\
             section=XShells\
             longtitle="XTerm: terminal emulator for X with Unicode support (
             title="UXTerm (ja_JP.UTF-8)"\
             command="sh -c 'LC_ALL=ja_JP.UTF-8 uxterm'"
   - /etc/menu/kterm:(強制複寫系統預設)<sup>10</sup>
            ?package(kterm):\
             needs="x11"
             section="XShells"\
             command="sh -c 'LC_ALL=ja_JP.eucJP PAGER=w3m /usr/X11R6/bin/kter
             title="Kanji Terminal"
             ?package(kterm):\
             needs="x11"
             section="XShells"\
             command="sh -c 'LANG=ja_JP.eucJP \
                       LC MESSAGES=en US.ISO-8859-1 PAGER=w3m /usr/X11R6/bin/k
             title="Kanji Terminal (bilingal)"
   - /etc/menu/mozilla-local:(加入新捷徑)<sup>11</sup>
             ?package(mozilla-browser):needs="x11" section="/" \
                     title=" Mozilla Navigator" command="mozilla-1.5" hints="W
<sup>9</sup>使用一個不會重複到其他套件名稱的名字。
```

<sup>10</sup>使用一個不會重複到這個套件的名字。

<sup>&</sup>lt;sup>11</sup>section="/"的斜線會使該項目加入到初始的選單中,title=" Mozilla Navigator"前頭的空白會使 該項目加入到列表最頂端。

- icon=/usr/share/pixmaps/mozilla.xpm - 從 root 帳號執行 update-menus。
- 加入底下的敘述到 ~/.muttrc (針對日文語系):

# UTF-8 support is not popular in popular Japanese EMACS environment # 7-bit encoding of iso-2022-jp is easier for everyone. # default encoding order = us-ascii --> iso-8859-1 --> iso-2022-jp set send\_charset="us-ascii:iso-8859-1:iso-2022-jp" set allow\_8bit=no

- 啓動 XIM kinput2 給 X 程式 (針對日文語系):
  - 加入 \*inputMethod: kinput2 和 KTerm\*VT100\*OpenIm: true 到您的 X 資源檔-~/.Xresources (Debian 應該會有某些方式自動完成這個動作)。
  - 有些程式 (例如 mlterm) 也允許您設定 \*inputMethod: 和其他資訊,並在執行時 動態載入(在 mlterm 按下**Ctrl-MouseButton-3**)。
- 執行 startx 來啓動 X 或任一個顯示管理器 (xdm, gdm, kdm, wdm, ...) (針對所有語系)。
- 在kterm下啓動一個相容日文的程式,例如 Vim 6, (x)emacs21, mc-4.5, mutt-1.4, ... (針對 日文語系)(Emacs 看起來是滿多人用的,雖然我不使用它。)
- 按下 Shift+Space 來啓動/關閉日文字元輸入模式(針對日文語系)。
- 在該語系相容的終端機下閱讀翻譯好的説明文件(針對所有語系)。

針對 CJK 語系支援,請參閱接下來的章節和 SuSE pages for CJK (http://www.suse.de/ ~mfabian/suse-cjk/suse-cjk.html)。

#### 9.7.10 其他的X 輸入法

還有許多其他的 X 輸入法支援的套件:

Language	LC_CTYPE	XIM server	XMODIFIERS	Start key
Japanese	ja_JP*	kinput2	"@im=kinput2"	Shift-Space
Korean	ko_KR*	ami	"@im=Ami"	Shift-Space
Chinese(T)	zh_TW.Big5	xcin	"@im=xcin-zh_TW.big5"	Ctrl-Space
Chinese(S)	zh_CN.GB2312	xcin	"@im=xcin-zh_CN.GB2312"	Ctrl-Space

日文輸入法 kinput2 程式是包在 kinput2-canna-wnn、kinput2-canna 和 kinput2-wnn 套件。理論上還需要辭典服務器如 canna 和 freewnn-jserver 才能使用。

## 9.7.11 X 終端機

還有許多 X 終端機支援 8-bit 編碼,安裝相關的字型就能正常顯示:

- xterm-X 終端機
- gnome-terminal-Gnome的 xterm
- konsole-KDE 的 xterm
- rxvt VT102 終端機 (lighter)
- aterm-Afterstep WM 的 VT102 終端機
- eterm Enlightment WM 的 VT102 終端機
- wterm-WindowMaker WM 的 VT102 終端機

xterm在 UTF-8 的編碼環境下就能支援雙位元組編碼(參閱 '支援 UTF-8 的 X 終端機' 158)。其他傳統編碼的終端機還在開發中(2003年時)。底下是已經支援傳統編碼的套間:

- aterm-ml Multi-lingual
- kterm Multi-lingual (Japanese, ...)
- rxvt-ml Multi-lingual
- wterm-ml Multi-lingual
- cxterm-big5 Chinese (Trad., Big5)
- cxterm-gb-Chinese (Simp., GB)
- cxterm-ks-Chinese(KS)
- cxterm-jis-Japanese
- hanterm-classic Korean (Hangul)
- hanterm-xf-Korean (Hangul)
- hztty Chinese (GB, Big5, zW/HZ)

使用 kterm 時(或其他終端機),您需要在選單上按下 Ctrl+滑鼠中鍵才能啓動 XIM。

#### 9.7.12 支援 UTF-8 的 X 終端機

支援 UTF-8 的終端機有 XFree86 4.x 版本的 xterm 套件裡的 uxterm 程式。它支援所有的語言。它是一個啓動後續程式使用 "UXTerm" X 資源集的一個 wrapper(參閱xterm(1))。

舉例還說,使用漂亮的大字型來顯示英文、俄文、日文、中文和韓文文字時,安裝對應的字型 套件後加入底下的設定到 ~/.Xresources 檔案。

```
! set large font
UXTerm*font: -misc-fixed-medium-r-normal-*-18-120-100-100-c-90-iso10646-1
! Use XIM for Japanese
*inputMethod: kinput2
```

然後執行 xrdb -merge ~/.Xresources 來更新 X 資源,就像 'X 資源' 137 這一節所提及的。

雖然大部分的終端機程式,例如 vim, mutt 和 emacs 已經相容 UTF-8(Woody-Sarge 之後的版本)了。但還是有少許的程式,如 mc 還不支援 UTF-8,可幸的是還有支援了 8-bit 編碼。如果 您編輯 7-bit ASCII 字元到未知編碼或是混用編碼的檔案,建議是使用無法辨識 locale 且支援 8-bit 編碼的編輯器是比較安全。

未來,每個人都會用到這些東西。參閱 The Unicode HOWTO (http://www.tldp.org/ HOWTO/Unicode-HOWTO.html)。

#### 9.7.13 支援 UTF-8 的 framebuffer 主控台

在 debian-installer 程式裡, bterm 是提供 FB 主控台支援 UTF-8 的套件。

## 9.7.14 locales 之外

當您第一次設定好國際語言環境之後,請使用 tasksel 或 aptitude 來安裝您的語言環境對 應的套件。該套件的安裝選單也適用於多國語言設定。如果您在安裝過程時遇到相依性問題 時,取消和避免安裝與系統衝突的套件。您或許需要使用 update-alternative 來回復指令 安裝前的狀態,因爲新安裝的套件有可能有更高的優先權覆蓋指令的連結。

使用 glibc 2.2 版的新版程式通常具備了國際化。所以特定的區域版本程式如 jvim 就不需要另 外安裝,因爲新的 vim 6.0 版本已經支援了它所提供的功能。事實上,還是有些小問題。由於 jvim 是針對日文輸入法支援所編譯的,所在在終端機上處理其他日文相關的文件,它還是很 值得推薦的:-)

應用程式有時需要忽略locale的設定才能適用於現有的工作環境。安裝 language-env 套件和使用其内建的指令 set-language-env 能很輕鬆地解決這些麻煩的步驟。

請參閱國際化相關文件, Introduction to i18n (http://www.debian.org/doc/manuals/ intro-i18n/)。該文件是針對開發者撰寫但也適用於系統管理者。

## 9.8 Multilingualization (m17n)

安裝 language-env 套件就能完成 'Localization (l10n)' 150 的設定步驟,該套件的目標就是將所有的語系視爲一種語言。這些套件也使用傳統的編碼方式處理文字檔案。在這個環境下,您不能混用法文和日文文字,因爲他們不相容 ISO-8859-1 和 EUC-IP。

使用 Gnome 和 KDE 程式在 UTF-8 的 locale 設定下,就能獲得多國語系的 UTF-8 桌面。 (Sarge)在這樣的環境下,您能混用英文、中文、俄文和日文字元在任何 UTF-8 相同的軟體 中。<sup>12</sup>

在這樣的系統下,新的多國語系輸入法建議使用 scim。由 scim 提供的輸入法只需按下 Ctrl-Space 就能啓動或關閉。滑鼠點選 SCIM 的控制視窗就能切換輸入轉換引擎。

vim offers the multilingualized environment and can handle both UTF-8 and conventionally encoded files (EUC-JP, ISO-8859-1, ...) when it is run under the UTF-8 console such as gnome-teminal. See vim help message with pressing [Esc] and typing :help mbyte.txt.

<sup>&</sup>lt;sup>12</sup>在多國語系環境下,language-env 套件並不那麼實用。

## Chapter 10

# 網路設定

這一章的重點在 Debian 的網路管理。請閱讀Net-HOWTO (http://www.tldp.org/ HOWTO/Net-HOWTO/index.html)了解一般 GNU/Linux 的網路設定。

爲了讓 Debian 的主機能夠成功連上網路,必須先設定好網路卡。

首重就是要確認核心支援的裝置,例如乙太網路卡,無線網路卡和數據機。您可能需要重新編譯核心並加入驅動程式模組'Debian下的Linux核心' 89來支援這些硬體。

以下說明如何設定網路裝置。這一章的內容主要是針對 Debian Sarge。大部分的內容不一定能相容早期的版本。

## 10.1 IP 網路基礎

一台 Debian主機會有多個 IP 位置。其硬體裝置有許多種,比如:

- Loopback: 10
- Ethernet: eth0, eth1,...
- Wi-Fi: wlan0, wlan1, wifi0,... <sup>1</sup>
- Token Ring: tr0, tr1,...
- PPP: ppp0, ppp1,...

還有其他不同的網路裝置,包含 SLIP、PLIP(串列和並列 IP)、控制網路流量的"shaper"裝置、frame 傳送、AX.25、X.25、ARCnet 以及 LocalTalk。

每一個連線到網路的裝置(或任何 IP 基礎的網路)將以唯一一個 32 bit 的 IP 位置來辨識。<sup>2</sup> IP 位 置可以分成網域位置和主機位置兩部份。將您的 IP 位置的網域位元全部設成1,而主機位元全 部設成0,則您會得到該網路遮罩。

以傳統上來說, IP 網路分成8、16、24 bits 長度的三個組別。3

<sup>&</sup>lt;sup>1</sup>注意,有些Wi-Fi裝置是抓用乙太網路裝置的名稱,只是參數的設定是針對Wi-Fi。這些參數是由iwconfig控制。

<sup>&</sup>lt;sup>2</sup>這是針對 IPV4 而言。IPV6是採用 128bit 定址,請參閱http://www.ipv6.org/。

<sup>&</sup>lt;sup>3</sup>這樣的系統缺乏彈性並會浪費許多網路位置,所以現在的 IPV4 網路是採用可變的長度來分組。

		IP addresses	S		net mask	le	ength
Class	А	1.0.0.0	_	126.255.255.255	255.0.0.0	=	/8
Class	В	128.0.0.0	_	191.255.255.255	255.255.0.0	=	/16
Class	С	192.0.0.0	_	223.255.255.255	255.255.255.0	=	/24

IP 位置不在這個範圍之内的是用在特殊環境上。

每一個類別的位置都給保留給本地網路(LANs)。這些位置保證不會影響到網路上的IP位置。(同理,如果主機被分配到這些地址的話,則該主機就無法直接存取網路,除非透過支援 proxy或是 Network Address Translation(NAT)功能才能夠連上網路。)底下的表格列出每個 類別的 IP 位置:

	network addresses			length	how many
Class	А	10.x.x.x		/ 8	1
Class	В	172.16.x.x -	172.31.x.x	/16	16
Class	С	192.168.0.x -	192.168.255.x	/24	256

IP 網路的第一個位置就是該網路本身,最後一個位置是這個網路的廣播位置。4 其餘的的 IP 位置就可以分配給其他該網域上的主機。通常IP位置的第一個與最後一個是用來當作網路通訊 閘。

路由表記載了核心如何傳送 IP 封包到目的端。底下説明一台位於本地端網路(LAN) IP 位置 爲 192.168.50.x/24 的 Debian 主機的路由表。另一台 192.168.50.1(也在 LAN 裡面) 的主機是 172.20.x.x/16 共用網路的路由器以及 192.168.50.254(也在 LAN 裡面)負責網際網路的路由器。

```
# route
Kernel IP routing table
                                          Flags Metric Ref Use Iface
Destination Gateway
                            Genmask
127.0.0.0
                            255.0.0.0
                                                0
                                                       0
                                                             2 10
                                          IJ
            *
                            255.255.255.0 U
192.168.50.0 *
                                                0
                                                       0
                                                           137 eth0
172.20.0.0
            192.168.50.1
                            255.255.0.0
                                          UG
                                                       0
                                                             7 eth0
                                                1
default
             192.168.50.254 0.0.0.0
                                          UG
                                                1
                                                       0
                                                            36 eth0
```

- 第一行説明傳遞的目的位置如果是 127.x.x.x, 則會透過 10(loopback)裝置路由。
- 第二行説明傳遞的目的位置如果是 LAN,則會透過 eth0 裝置路由。
- 第三行説明傳遞的目的位置如果是共用網路,則會透過 eth0 裝置路由,最後傳送到 192.168.50.1 通訊閘。
- 第四行説明傳遞的目的位置如果是網際網路,則會透過 eth0 裝置路由,最後傳送到 192.168.50.254 通訊閘。

路由表的 IP 位置也以名稱表示,這些名稱由/etc/networks或 resolver C函式來取得。

除了路由支外,核心也能實現 network address translation(NAT)、流量控制以及封包過濾。

請 閱 讀 Net-HOWTO (http://www.tldp.org/HOWTO/Net-HOWTO/index. html) 以 及 other networking HOWTOs (http://www.tldp.org/HOWTO/ Networking-Overview-HOWTO.html)了解背後的運作原理。

<sup>&</sup>lt;sup>4</sup>網域的計算方式是將 IP 位置與網路遮罩進行 AND 運算。而廣播位置的計算是先網路遮罩經過一補數計算再 與網域進行 OR 運算。
### 10.2 低階網路設定

傳統的 GNU/Linux 系統設定網路的工具是 ifconfig and route,這兩個程式可以從 net-tools 套件取得。目前有新的工具: ip 可以取代這兩個程式的功能,您可以從 iproute 套件取得。ip 程式需搭配 Linux 2.2 或更新的核心且比舊有的程式功能更強。即使如此,這些傳統的設定工具還是相當實用且大家也比較熟悉。

#### 10.2.1 低階網路設定 - 使用 if config 和 route

以下的説明示範如何改變eth0的 IP 從 192.168.0.3 到 192.168.0.111 以及透 過192.168.0.1 路由到 10.0.0 這個網域。執行 ifconfig 和 route 時不加入網路 裝置會顯示目前的網路裝置和路由表狀態。

```
# ifconfig
```

- eth0 Link encap:Ethernet HWaddr 08:00:46:7A:02:B0
  inet addr:192.168.0.3 Bcast:192.168.255.255 Mask:255.255.0.0
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:23363 errors:0 dropped:0 overruns:0 frame:0
  TX packets:21798 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:100
  RX bytes:13479541 (12.8 MiB) TX bytes:20262643 (19.3 MiB)
  Interrupt:9
- Link encap:Local Loopback lo inet addr:127.0.0.1 Mask:255.0.0.0 UP LOOPBACK RUNNING MTU:16436 Metric:1 RX packets:230172 errors:0 dropped:0 overruns:0 frame:0 TX packets:230172 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:22685256 (21.6 MiB) TX bytes:22685256 (21.6 MiB) # route Kernel IP routing table Destination Gateway Flags Metric Ref Use Iface Genmask 192.168.0.0 \* 255.255.0.0 U 0 0 0 eth0 192.168.0.1 255.255.255.255 UG 0 0 default. 0 eth0

首先,我們先關閉 eth0。

# ifconfig eth0 inet down
# ifconfig
lo Link encap:Local Loopback
... (no more eth0 entry)
# route
... (no more routing table entries)

接下來我們啓動 eth0 並給予新 IP 位置和路由。

結果是:

```
# ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:46:7A:02:B0
    inet addr:192.168.0.111 Bcast:192.168.0.255 Mask:255.255.255.0
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    . . .
lo
   Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    . . .
# route
Kernel IP routing table
Destination Gateway
                     Genmask Flags Metric Ref Use Iface
192.168.0.0 *
                       255.255.255.0 U 0
                                                0 0 eth0
10.0.0.0 192.168.0.1 255.0.0.0
                                 UG O
                                                0
                                                      0 eth0
```

請參閱 if config(8) 和 route(8) 取得更多資訊。

#### 10.2.2 低階網路設定-ip

ip 與 if config和 route 有同樣功能的指令如下:

- ip link show
- ip route list
- ip link set eth0 down
- ip addr del dev eth0 local 192.168.0.3
- ip addr add dev eth0 local 192.168.0.111/24 broadcast 192.168.0.255
- ip link set eth0 up
- ip route add dev eth0 to 10.0.0.0/8 src 192.168.0.111 via 192.168.0.1

執行 ip 指令時加入 help 參數會顯示指令的用法。 例如, ip link help 代表:

Usage: ip link set DEVICE { up | down | arp { on | off } | dynamic { on | off } | multicast { on | off } | txqueuelen PACKETS | name NEWNAME | address LLADDR | broadcast LLADDR | mtu MTU } ip link show [ DEVICE ]

請參閱 ip(8)。

#### 10.2.3 設定 Wi-Fi 裝置

想要設定 Wi-Fi 裝置,除了ifconfig 或 ip之外,您還需要安裝wireless-tools並使用内 附的iwconfig程式。

請參閱 iwconfig(8)。

#### 10.2.4 設定 PPP 装置

如果您透過 modem 來撥接上網的話,這樣的連線被稱為 Point-to-Point (PPP) 協定。這樣連線作業的網路裝置為 ppp0, ppp1 等等。

一個 PPP 裝置是由 pppd 這個 PPP 服務所管理,您可以在 ppp 套件中找到該程式。因此,設定 PPP 裝置也就是設定 pppd 的意思。

#### 手動設定 pppd

一個網路連線的建立需要完成以下一連串的手續:開啓通訊埠(通常是串列埠),透過通訊裝置送出指令(通常是數據機),撥出電話號碼,發送端的身分必須給外界的 PPP 服務認證,由核心建立 PPP 裝置並設定路由表,使得資料可以透過該線路傳遞。pppd 能完成以上所有的動作,所以會有一堆設定參數。關於參數的説明請參閱 pppd(8)。

在Debian系統中,全域選項是設定在/etc/ppp/options。使用者格式選項是設定在~ /.ppprc。必需依賴通訊埠來使用的選項放置在/etc/ppp/options.portname。舉例來 說,假設你有兩台數據機—一固定式的 Lucent LT modem 從/dev/LT-modem來存取資料和 一外部的 modem 從/dev/ttyS0來存取資料。建立下列這兩個選項檔案。

```
# cat > /etc/ppp/options.LT-modem <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-LT-modem"
EOF
# cat > /etc/ppp/options.ttyS0 <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-ttyS0"
EOF</pre>
```

這些涉及到下列多人線上交談系統的指令碼。首先,/etc/chatscripts/setup-LT-modem.

ABORT ERROR '' ATZ OK 'ATW2X2 S7=70 S11=55' OK AT

第二,/etc/chatscripts/setup-ttySO.

```
ABORT ERROR
'' ATZ
OK 'ATL1M1Q0V1W2X4&C1&D2 S6=4 S7=70 S11=55 S95=63 S109=1 +FCLASS=0'
OK AT
```

當然,這一些檔案的內容必需依賴你的硬體。

選項也可以跟參數一樣傳遞給 pppd。

在Debian pppd 通常使用pon此命令來啓動。當 pon 被它的第一個參數名字來使用時,在 /etc/ppp/peers/的一個選項檔案也同時被pppd讀取。<sup>5</sup>該檔是您針對特定的網點服務所 設定的地方,舉例來說,一個特定的網路服務提供者(ISP).

假設,舉例來說你從Amsterdam和Den Haag之間來溝通。在每一個城市你使用了兩個 ISP services—Planet 和 KPN.首先對每一個ISP建立一個基本的設定檔案。

```
# cat > /etc/ppp/peers/KPN <<EOF</pre>
remotename KPN
noauth
user kpn
noipdefault
ipparam KPN
EOF
# cat > /etc/ppp/peers/Planet <<EOF</pre>
remotename Planet
auth
user user35790planet.nl
noipdefault
mru 1000
mtu 1000
ipparam Planet
EOF
```

這一些檔案從兩個不同的ISP來設定選項。對這兩個ISP來說常見的選項可以被放置在/etc/ppp/options或當一種特殊介面的選項檔案被佔用時。

現在對每一個城市的ISP建立選項文件。在我們的例子中連結到不同地區ISP的差別只在於對話script是否需要。(對話script的不同,因爲區域接收資料的電話號碼是不同的。)

<sup>5</sup>這一個設定檔案包括使用這一個 call 選項。

```
# cat > /etc/ppp/peers/KPN-Amsterdam <<EOF</pre>
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-Amsterdam"
file /etc/ppp/peers/KPN
EOF
# cat > /etc/ppp/peers/KPN-DenHaag <<EOF</pre>
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-DenHaag"
file /etc/ppp/peers/KPN
EOF
# cat > /etc/ppp/peers/Planet-Amsterdam <<EOF</pre>
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-Amsterdam"
file /etc/ppp/peers/Planet
EOF
# cat > /etc/ppp/peers/Planet-DenHaag <<EOF</pre>
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-DenHaag"
file /etc/ppp/peers/Planet
EOF
```

file 指令顯示之前各包含了一個選項檔案。connect 指令詳細地說明了用來建立連線的pppd 命令。通常使用這一個 chat 程式來做,使對話script來適應ISP。這裡是給Den Haag的對 話script;和給Amsterdam的對話script除了電話號碼以外可能極爲相似或者當ISP從一個公司 到另外一個公司所提供的服務有可能是不同的。

# cat > /etc/chatscripts/KPN-DenHaag <<EOF</pre> ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE ABORT 'NO DIALTONE' ABORT 'NO DIAL TONE' ABORT 'NO ANSWER' ABORT ERROR OK-AT-OK ATDT 0676012321 CONNECT \d\c EOF # cat > /etc/chatscripts/Planet-DenHaag <<EOF</pre> ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE ABORT 'NO DIALTONE' ABORT 'NO DIAL TONE' ABORT 'NO ANSWER' ABORT ERROR OK-AT-OK ATDT 0676002505 CONNECT \d\c EOF

爲了能連接上這些 ISP,你需要的用户名稱和密碼在被要求時以便讓 pppd 可以提供資料。 這一資訊不是被儲存在/etc/ppp/pap-secrets (假使 PAP 規則使用中)就是在 /etc/ppp /chap-secrets (假使 CHAP 規則使用中)。雖然 CHAP 是相當安全的, PAP 依然是相當 廣泛地被使用。因爲這一些檔案包含了"secret",所以群組和其他用户並不被允許讀取或是寫 入。這些檔案的格式請參閱pppd(8) 的說明。"secret"(第三格)是由用户端名稱(第一格)和/或 伺服器名稱(第二格)來搜尋。當連線到一個ISP時通常是不知道那一個伺服器的名稱,所以提供 一個用户端名稱來取代;這在 peers/KPN 和 peers/Planet 檔案中的 user 行就能完成。

# client name	server	name	secret
kpn	*		kpn
user35790planet.nl	*		myfavoritepet

See /usr/share/doc/ppp/README.Debian.gz for more information.

#### 使用 pppconfig 來設定 pppd

一個快速的方法來設定 pppd 是使用pppconfig 這一個用相同名稱的套件。pppconfig 會透 過選單介面來詢問用户問題,之後設定檔案的方式就如同上述。

#### 使用wvdial來設定 PPP 裝置介面

另一個使用 pppd 不同的方式是從wvdial 來執行它, wvdial包裝在 wvdial 套件裡。取代了 pppd 用 chat 來撥接和協調連線的動作, wvdial完成了撥接和初始協調的動作, 剩下的就交 給pppd。只要給予電話號碼,使用者名稱和密碼, wvdial 大部份都能成功地完成連線動作。

# 10.3 對主機命名

#### 10.3.1 主機名稱

hostname-系統主機名稱是由核心所維護的。初始的script /etc/init.d/hostname.sh 會 在開機時設定系統主機名稱(使用 hostname 這個命令),名稱儲存在 /etc/hostname。這個 檔案應該 只 包含系統主機名稱,而不是一個FQDN。

執行 hostname (不帶任何參數)來顯示目前的主機名稱。

#### 10.3.2 郵件名稱

一個主機的郵件名稱是郵件相關的程式用來確認主機的。 /etc/mailname 包含了該名稱並以 新空行結尾。郵件名稱通常是一個可解折主機IP位置的FQDN。參閱 mailname(5)。

電郵的收件者發現郵件標頭的 From:時,它的值是依據您的 Debian 主機的郵件用户代理(MUA)和郵件傳送代理(MTA)的設定。假設一本地端用户 foo 從郵件名稱為 myhost.dom 的主機傳送郵件時。送出的電郵標頭 From: 會變成:

- "From: foo@myhost.dom" if the MUA has no From: header set;
- "From: bar@myhost.dom" if the MUA has "From: bar" set;

• "From: bar@bogus.dom" if the MUA has "From: bar@bogus.dom" set. 就算 MUA 有 From: 標頭, MTA 還是會加入 "Sender: foo@herman.dom" 標頭來指定真實 的來源。

當然,任何複雜的 MTA 在執行位址重寫時,如同'在 Exim 下設定不存在的電郵位址的垃圾桶' 147 和'在 Exim 下設定寄出郵件的位址重寫'148 的討論,收件者顯示的郵件地址可以任意更 改。

### 10.4 網域名稱服務 (DNS)

主機由網域名稱和IP地址所查詢。DNS用途是把網域名稱和IP位址以及主機的其他特性聯繫起來的一個客服端-伺服器系統。GNU C Library resolver (3) 也可從檔案中查詢IP位址或查閱 Network Information Services (NIS)。

有些軟體(像是GNOME)希望系統主機名稱是一個可解折IP位置的FQDN。然而,這並不洽當,因爲系統主機名稱和網域名稱兩者是截然不同的。爲了支援這樣的軟體,必須確定系統主機名稱是可被解折的。常見的解法是加入系統主機名稱和IP位置的敘述到/etc/hosts。如果您的系統有固定IP,就使用那一組IP位置,否組請用127.0.1.1這個IP位置。

127.0.0.1 localhost 127.0.1.1 uranus

執行hostname --fgdn來檢查您的FQDN是解折成那一個IP位置。

#### 10.4.1 resolver

查明IP位址和一特殊網域名稱的聯繫就是 resolver 的工作。大部份常用的 resolver 是 GNU C Library 中的 resolver 所提供的功能(resolver(3))。另一個是由libfiredns套件所提供的 FireDNS resolver。除了這兩個外,還有其它同功能的套件喔。

GNU LIBC 的 resolver 如何控制 /etc/nsswitch.conf 設定檔中含有 hosts 那一行來 解決名稱查詢呢?該行列出查詢名稱的服務:例如 dns, files, nis, nisplus。 參閱 nsswitch.conf(5)。在使用 files 的情況下, resolver 也是由 /etc/hosts 設定檔控制。 參閱 hosts(5)。

上述的檔案是固定的且可用您喜愛的編輯器修改。

#### 10.4.2 管理名稱伺服器的資訊 - resolvconf

resolvconf提供一個框架來動態管理可使用的名稱伺服器資訊。它解決了長久以來維護 resolver 和 DNS 快取使用的動態名稱伺服器的列表問題。Resolvconf 扮演著各個控制網路裝 置介面,提供名稱伺服器資訊的程式或需要名稱伺服器資訊的軟體的媒介。

resolvconf被設計成無須任何手動設定就能工作。但該套件還相當新,可能需要部份的手動 調整才能正常工作。如果您曾改造過套件使它會更新/etc/resolv.conf的話,您將需要取 消修改的部份。更多訊息請參閱/usr/share/doc/resolvconf/README.gz。

### 10.4.3 快取已查閲的名稱 - nscd, dnsmasq, pdnsd, bind9

如果你的名稱伺服器反應緩慢你可能想要使用nscd 來快取使用libc6 resolver 所查詢的結果。

如果你想要在你的區域網路快取其他主機名稱,你可能要執行一個快取前置名稱伺服器,就像 dnsmasg 或 pdnsd。

如果你希望也可以使用 bind9 套件的 named程式當一個快取前置名稱伺服器。雖然這是一個 重量級的程式,除非你需要他先進的功能,否則你使用先前提及的套件是比較好的。

所有這些的套件皆與 resolvconf 運作良好。

#### 10.4.4 提供網路名稱服務-bind

如果你需要提供可靠的名稱服務給一個網域,你需要一個完全成熟的名稱伺服器,例如套件 bind9中的 named。

如果你安裝 bind9 你也應該安裝 dnsutils. 你可能也需要安裝這一些公用的套件: bind9-host; dns-browse; dnscvsutil; nslint. 你可能也需要安裝這個使用說明套件: bind9-doc. 你可能也需要安裝這一些開發套件: libbind-dev; libnet-dns-perl. 如果 你安裝界面是使用 DHCP 你可能找到這一有用的套件: dhcp-dns.

安裝 bind9 或 dpkg-reconfigure 它來作基本的設定。設定包含編輯 named.conf。在 Debian 裡,該檔案在 /etc/bind/ 找到且主要被使用來定義基本的 DNS 區域;它包含了 兩個其他文件:named.conf.local,用來定義區域地區,和named.conf.options,用 來設定選項。(後者是靠著 resolvconf 來產生 /var/run/bind/named.options,除了 forwarders 說明是目前可得到的非區域的名稱伺服器的目錄之外,其餘的都是和原來的 相同。要利用這功能,改變named.conf 中的 include 這一行,使得它包含了 /var/run /bind/named.options。參閱 '管理名稱伺服器的資訊 - resolvconf' 170。)

在 named.conf\* 目錄下的沒有全部路徑名稱的資料庫檔案會被儲存在 /var/cache/bind /。這是最合適儲存由named 產生檔案的位置:舉例來說, zone 的資料庫檔案為第一,而 daemon為第二。在named.conf的資料必須依靠完整路徑才能找到/etc/bind/目錄下的固 定資料庫檔案。詳細訊息請參閱 /usr/share/doc/bind9/README.Debian.gz。

# 10.5 安裝網路界面使用 DHCP

低階的網路安裝界面靠著 Dynamic Host Configuration Protocol (DHCP)可以被自動安裝。你的防火牆或路由器或寬頻的 ISP 可能用此方法來配置IP位址和其他參數。

要做到這項工作你必順安裝下列的套件之一:

- dhcp3-client (version 3, Internet Software Consortium)
- dhcpcd (Yoichi Hariguchi and Sergei Viznyuk)
- pump (Red Hat)

pump 是一套簡單且廣泛使用的套件。 dhcp3-client 是複雜的但常安裝的。 6

## 10.6 Debian 的高階網路設定

#### 10.6.1 高階網路設定-使用ifupdown

為了要使網路安裝更加容易, Debian 提供了一個標準的高階網路工具由 ifup 和and ifdown 程式和/etc/network/interfaces 檔案所組成。<sup>7</sup>如果你選擇去使用 ifupdown 來做你的 網路設定,那麼你應該不使用低階的命令。這也表示你不應該使用其它高階結構工具,例如會 呼叫低階結構工具的whereami, divine, intuitively等等。這 ifupdown 程式被設計的 目地是要被獨立地使用來安裝和反安裝網路界面。

要更新介面設定請執行:

- # ifdown eth0
  # editor /etc/network/interfaces # 依您的需要調整
- # ifup eth0

更多資訊請參閱 interfaces(5), /usr/share/doc/ifupdown/examples/ network-interfaces.gz和ifup(8)。

#### 以固定IP位址來安裝一個界面

假設你要安裝一乙太網路界面以致它有一固定的 192.168.0.111 IP位址。這位址從 192.168.0 開始,所以它一定在 LAN 上。更深一步地假設, 192.168.0.1 是 LAN 連 上網際網路的通訊閘位址。編輯 /etc/network/interfaces 以致於它包含了像這樣的資料:

iface eth0 inet static

<sup>&</sup>lt;sup>6</sup>在 2004 年四月也有一個套件 dhcp-client 可以利用。這包含了 ISC DHCP 用户端的第二個版本。這已經 被目前套件 dhcp3-client 的第三版本所取代。維護人員計劃在 Sarge 釋放之後將 dhcp3-client 重新命名為 dhcp-client 。 確認你沒有安裝 dhcp-client 的測試版本。ifupdown 並不和他們一起運作。

<sup>&</sup>lt;sup>7</sup>/etc/network/interfaces的格式與現行的 ifupdown 版本和早先的 Potato 版本的套件是不能並存的。 這 ifupdown 在後安裝的 script 在必要時應該自動的升級檔案。無論如何,去檢查有改變的檔案這是一個好主意。

address 192.168.0.111 netmask 255.255.255.0 gateway 192.168.0.1

你可以設定介面的其他部份或是執行其他動作,在界面被帶起之後或在它被帶下之前,靠著是 "up"和"down"那一行適當的命令説明。

```
iface eth0 inet static
    address 192.168.0.111
    netmask 255.255.255.0
    gateway 192.168.0.1
    up route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFAC
    down route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFAC
    up echo Interface $IFACE going up | /usr/bin/logger -t ifup
    down echo Interface $IFACE Going down | /usr/bin/logger -t ifdown
```

二者擇一地,命令可以在 /etc/network/if-up.d 和 /etc/network/if-down.d 資料夾 被插入在scripts之内。這樣的 scripts 也可以執行延伸的選項。參閱 interfaces(5) 來了解 細節。舉例來説, resolvconf 套件納入的 script 在界面帶起時允許你增加選項來說明被包含 在 /etc/resolv.conf 的DNS資訊:

```
iface eth0 inet static
    address 192.168.0.111
    netmask 255.255.255.0
    gateway 192.168.0.1
    dns-search somedomain.org
    dns-nameservers 195.238.2.21 195.238.2.22
```

在 dns-search 的 somedomain.org 參數選項符合在 resolv.conf(5) 的 search 參 數選項。 在 dns-nameservers 的 195.238.2.21 和 195.238.2.22 參數選項符合這 nameserver 參數選項。 其他識別的選項是 dns-domain 和 dns-sortlist。參閱'管理 名稱伺服器的資訊-resolvconf'170。

#### 使用DHCP來安裝裝置界面

使用DHCP來安裝一個界面裝置介面,請編輯/etc/network/interfaces,包含了像這樣的敘述:

iface eth0 inet dhcp

爲了要使這功能運作你必須安裝一個在'安裝網路界面使用 DHCP' 171 所提及的DHCP用户端 程式。

#### 安裝 Wi-Fi 裝置介面

wireless-tools 這個套件包含了一個 hook script /etc/network/if-pre-up.d /wireless-tools 使得界面被帶起之前順利安裝 Wi-Fi (802.11a/b/g) 硬體變成可能。使用 iwconfig 這一程式來完成設定;參閱 iwconfig(8)。對 iwconfig 的每一可能命令參數,您可以在 /etc/network/interfaces 包含一個選項,像是 "wireless-"前綴參數一樣來命名。舉例來說,在使用 DHCP 帶起 eth0 之前,設定 eth0 的 ESSID 到myessid 和加密 金鑰到 123456789e,編輯/etc/network/interfaces 以致於它包含了像這樣的敘述:

```
iface eth0 inet dhcp
wireless-essid myessid
wireless-key 123456789e
```

注意!如果你執行 waproamd 來設定這裝置介面,你不應該使用這個方法來設定 ESSID 和金鑰。在 ifup 執行時, waproamd 已經設定了 ESSID 和金鑰。參閱 '觸發網路設定 – 使用 waproamd' 180。

#### 設定 PPP 介面

ifup 和 ifdown 程式使用 pon 和 poff來增加和移除 PPP 界面,所以首先閱讀 '設定 PPP 裝置' 165。

假設你已經設定了 PPP 和 myisp 一起運作。編輯 /etc/network/interfaces 以致於它包含了像這樣的敘述:

iface ppp0 inet ppp provider myisp

如同這樣的敘述, ifup ppp0 會完成

pon myisp

遺憾的是目前無法提供額外的 pppd 選項到 /etc/network/interfaces 中的 ppp 敘述。8

目前使用 ifupdown 來執行 PPP 界面的輔助設定是不可能的。因為 pon 在 pppd 完成連結之前就存在了,在 PPP 界面準備使用之前, ifup 會預先執行 up scripts。 直到這個 bug<sup>9</sup> 被修 正前,還是要額外設定 /etc/ppp/ip-up 或 /etc/ppp/ip-up.d/。

<sup>&</sup>lt;sup>8</sup>參閱 bug #196877 (http://bugs.debian.org/196877)。

<sup>&</sup>lt;sup>9</sup>參閱 bug #127786 (http://bugs.debian.org/127786)。

#### 設定 PPPoE 裝置界面

許多寬頻網路供應商 (ISPs) 使用 PPP 協調連線,即使用者的機器透過 Ethernet 和/或 ATM 網路來連接他們。完成這樣的功能需依靠封裝乙太網路的 frame 的 PPP 串流技術,也就是 PPP over Ethernet (PPPoE)。 假設你的 ISP 是叫做 myisp。首先對 myisp 設定 PPP 和 PPPoE。 最容易做到這的方法是從主控台安裝 pppoeconf 套件且執行 pppoeconf。然後編輯 /etc /network/interfaces 以致於它包含了像這樣的敘述:

```
iface eth0 inet ppp
provider myisp
```

有時候最大傳輸單位 Maximum Transmit Unit (MTU) 決定於 Digital Subscriber Line (DSL) 之上的 PPPoE。參閱 DSL-HOWTO (http://www.tldp.org/HOWTO/DSL-HOWTO/) 來了解細節。

注意!如果你的寬頻數據機包含路由器功能,那麼數據機/路由器本身處理 PPPoE 連線並當作 一條簡單的乙太網路通訊閘連向網際網路的路徑出現在 LAN 周邊。

#### 爲通訊閘設定多個乙太網路裝置介面

假設 eth0 用一個 DHCP-configured IP 位址連接到網際網路且 eth1 用一個固定的 IP 位址 192.168.1.1 連接到 LAN 。編輯 /etc/network/interfaces 以致於它包含了像這樣的 敘述:

iface eth0 inet dhcp iface eth1 inet static address 192.168.1.1 netmask 255.255.255.0

如果你如'建立通訊閘路由器' 184 所描述的去啓動在主機上的 NAT, 那麼你可以分享網際網路 連接在 LAN 上的所有主機。

#### 設定虛擬裝置介面

使用虛擬裝置介面,你可以安裝一張乙太網路卡去成為好幾個 IP 子網面的介面。舉例來說,假設你的主機是在 LAN 網路上 192.168.0.x/24。你想要讓主機連線到網際網路,並用原本的網路卡透過 DHCP 機制取得真實 IP 位址。編輯 /etc/network/interfaces 以致於它包含了像這一些敘述:

iface eth0 inet static address 192.168.0.1 netmask 255.255.255.0 network 192.168.0.0 broadcast 192.168.0.255

iface eth0:0 inet dhcp

這 eth0:0 界面是一個虛擬的界面。當它被帶起時,它的根源就是 eth0。

#### 10.6.2 高階網路設定-ifupdow邏輯裝置定義

對讀者而言,了解下列文章中的physical interface 和 logical interface 之間的差異是很重要 的。<sup>10</sup>一個 physical 裝置界面是我們稱呼爲"裝置界面",核心命名爲 eth0, eth1, ppp0 ,諸如此類的。一 logical 裝置界面是一套可以指定 physical 裝置界面的可變參數值。當妳 在閱讀的同時,如果發現有困惑,用這個述敘 "configured with interface profile X" 來取代 "configured as logical interface X"。

那些定義在 /etc/network/interfaces的iface, 實際上是定義邏輯裝置界面而不是 physical 裝置界面。<sup>11</sup> 如果你從不想要去重安裝你的界面, 那麼你可以忽略這個細節, 因 爲 physical 裝置介面 foo 預設會被設定成邏輯裝置介面 foo。

然而,假設你的電腦是你在家裡和工作之間運輸的筆記型電腦。當你的電腦連線到公司的網路 或是到你的家用 LAN 時,相應地你需要設定 eth0。

首先定義兩個邏輯裝置界面 home 和 work(取代 eth0,就像我們先前做的),針對家用網路和 公司網路各別地設定裝置介面。

iface home inet static address 192.168.0.123 netmask 255.255.255.0 gateway 192.168.0.1

iface work inet static address 81.201.3.123 netmask 255.255.0.0 gateway 81.201.1.1

透過適當的設定後,只要在命令列上指定後,physical 裝置界面 eth0 就能在家庭網路上被帶起:

# ifup eth0=home

針對公司網路重新設定 eth0 只要執行這些命令:

<sup>&</sup>lt;sup>10</sup>這專門術語是使用在 ifupdown 套件中的使用說明。

<sup>&</sup>lt;sup>11</sup>注意,命名含有 auto字串的那一行上的裝置界面必須是 physical 裝置界面,而不是邏輯裝置界面。

# ifdown eth0

# ifup eth0=work

注意,如果interfaces設定檔內容如上面寫所的,即使單獨執行 ifup eth0 也不會帶起 eth0。這理由是因爲, ifup 使用 physical 裝置界面名稱來當作預設的邏輯裝置介面名稱, 然 而我們的例子中沒有定義 eth0 這個邏輯裝置介面。

#### 10.6.3 神奇的網路設定方式

裝置介面名稱可以在執行 ifup 時對應成其他名稱。至於對應成什麼名稱則是依照情況判定。因此ifup能設定在帶起實體裝置介面時能對應成多個預定義的邏輯裝置。

邏輯裝置介面名稱發生對應的情況如下:

- 如果執行ifup時沒指定邏輯裝置介面,則實體裝置介面名稱爲初始的邏輯裝置介面。
- 如果邏輯裝置介面符合 mapping 敘述的 glob-pattern,則對應到新產生的邏輯裝置介面。這些動作在順序的 mapping 敘述就已經完成了。
- 如果最終的的邏輯裝置介面名稱是/etc/network/interfaces檔案中的一個邏輯裝置 介面定義的標誌,則實體裝置介面被帶起成邏輯裝置介面。不然 ifup 會印出 "Ignoring unknown interface" 的訊息並離開。

mapping 敘述的文法是:

mapping glob-pattern
 script script-name
 [map script input]

mapping 敘述命名的 script 會加入 physical 裝置介面爲參數執行,而剩下的敘述中的 "map"(不包含"map"字串) 行的内容會導入到它的標準輸入。該 script 在離開前會印出對 應到他的標準輸出的結果。

舉例來説,底下的 mapping 敘述會促使 ifup 帶起 eth0 裝置介面並當成 home 邏輯裝置介面。

mapping eth0
 script /usr/local/sbin/echo-home

/usr/local/sbin/echo-home 的内容是:

#!/bin/sh
echo home

因爲對使用 script 完成對應,所以自動選擇邏輯裝置介面是可能的 — 基於幾個測試的結果。參 閱 '使用 guessnet 選擇邏輯裝置介面' 177 中的範例。

#### 使用 guessnet 選擇邏輯裝置介面

安裝 quessnet 套件並加入以下的敘述到 /etc/network/interfaces:

```
mapping eth0
    script guessnet-ifupdown
    map home
    map work
```

現在起,當妳執行 ifup eth0 時, guessnet 會檢查 eth0 是否被帶起成 home 或 work。 它使用儲存在邏輯裝置介面的定義來完成檢查的動作。

#### 10.6.4 利用 laptop-net 自動設定網路

laptop-net套件使用不同的方法來自動設定網路。Laptop-net 不採用 ifupdown 的邏輯裝置 反而改用它内建的系統設定 的"方案"以及系統"偏好"。但是,Laptop-net 仍然使用 ifup 和 ifdown 來設定硬體裝置。安裝 laptop-net-doc 套件來閱讀詳盡説明的文件。

#### 10.6.5 利用network-manager自動設定網路

network-manager是由Fedora開發者所發明的並包裝到Ubuntu上。它未來可能會納入Debian中,而同時也表示ifupdown及其它軟體就廢棄不用了。

# 10.7 處理核心的非一致性裝置命名規則

eth0, eth1 等裝置名稱是由核心指定的,而核心建立裝置的順序是依照這些名稱。在開機時 偵測到介面卡後,通常是以同樣的順序來偵測,所以指定的裝置名稱也是依照同樣的順序,然 而,該規則無法套用在可熱插拔的裝置上。在不同的情況下,這些裝置被偵測到後,核心可以 用任何順序或規則來指定裝置節點。

因爲這樣的關係,系統上使用可熱插拔的網路卡將不會採用 /etc/network/interfaces定 義好的邏輯裝置,如 eth0,eth1 等預設的裝置對映。取而代之的是您必須針對這邏輯裝置給 定一個別的名稱並使用下列兩者其一的方式來控制那一個介面卡指定成那一個邏輯裝置。

«««< gateway.sgml 這個問題只要使用目前的udev套件版本和設定就能解決了。 »»» 1.16 使用 nameif 工具(在 net-tools 套件中)或更具彈性的 ifrename 工具(在 ifrename 套件中)使 得核心依照介面卡的屬性來指定裝置名稱。採用這種命名規則的話,介面卡就能推測是使用那 一個實體裝置名稱。

另一個辦法是採用 ifup 的對應技術使得介面卡在被帶起時能依據部份屬性來找到對應的邏輯 裝置介面。

假設,舉例説明,您有兩個不同的網路卡,對應的網路分別為 net1 和 net2。/usr/share /doc/ifupdown/examples/目錄下包含了對應 script,依照 Media Access Controller address (MAC 位址) 來選擇介面卡的邏輯裝置介面。先安裝該 script 到適當的目錄。 # install -m770 /usr/share/doc/ifupdown/examples/get-mac-address.sh \
 /usr/local/sbin/

然後增加類似下列的敘述到 /etc/network/interfaces:

```
mapping eth0
    script /usr/local/sbin/get-mac-address.sh
    map 02:23:45:3C:45:3C net1
    map 00:A3:03:63:26:93 net2
```

參閱 '多階段(Multi-stage)對應' 181 取得更進階的範例。

不管採用那一種方式,最常見的是用 MAC 位址來辨認介面卡。 «««< gateway.sgml -> ====== -> »»»> 1.16

# 10.8 觸發(Triggering)網路設定

我們已經了解裝置介面是如何設定或重新設定。這些動作需要在適當時機就完成。

傳統上,網路在開機時會經由 /etc/rcS.d/S40networking 這個 initscript 設定完成, 極少數的機會重新設定。其他網路服務會在陸續啓動。反過來說,在關機或重開機的時候, initscript 執行的時機剛好相反。

現在,GUN和 Linux 正朝著支援動態更換硬體和處理突發事件產生的趨勢。首先支援的是可 熱(線上)抽換的 PCMCIA 卡。目前的 hotplug 已經支援更多可抽換的週邊硬體,即使是電腦 運行中。這也包含了網路硬體。注意,當您移除可熱(線上)抽換的硬體時,牽涉到硬體的服務 必須先停止。這表示這類型的服務必須從 System V init 系統移除並歸屬在 ifupdown 的控制 下。

例如,支援控制 foo 服務的 initscript:/etc/init.d/foo 仰賴於動態設定的網路介面 eth0。

首先,從 init 系統中移除 foo。如果您使用 sysv-rc init 系統的話,請完成下列事項。
 12

```
# rm /etc/rc[2345].d/S??foo
```

• 然後將 foo 加入ifupdown的控管下,新增up 和 down參數到/etc/network /interfaces檔案的 eth0敘述。該檔案會呼叫 foo initscript:

iface eth0 inet dhcp up /etc/init.d/foo start down /etc/init.d/foo stop

#### 10.8.1 在開機時觸發(trigging)網路設定

在啓動 /etc/rcS.d/S40networking init script 時會執行 ifup -a 這個指令。這會帶起 /etc/network/interfaces檔案中所有auto敘述的硬體裝置介面。

<sup>&</sup>lt;sup>12</sup>注意,這個動作忽略了 "stop"軟連結(/etc/rc?.d/K??foo)。更多訊息請參閱 'Runlevels' 18。

這表示使用動態方式來設定網路會比較適當。當支援動態更動硬體的技術成熟後,這會是最簡單的方式來轉化固定的硬體當成動態裝置處理。啓動裝置就能當成是另一個熱插拔的事件。(參閱 / 觸發網路設定 – 使用hotplug' 179。)

儘管如此,大部份的情況下,至少10loopback裝置要在開機時被帶起。所以,確定/etc/network/interfaces中有包含底下的敘述。

auto lo

iface lo inet loopback

如果妳想讓其他硬體裝置在開機時也被帶起,請加入 auto 敘述吧。絕對不要在 PCMCIA 介面上加入auto敘述。PCMCIA 的管理程式: cardmgr會在 /etc/rcS.d/S40networking執行後啓動。

#### 10.8.2 觸發網路設定 – 使用hotplug

安裝 hotplug 套件以尋求 熱抽換的支援。

你可以在開機時或是安裝介面卡(例如 PCMCIA 卡)到系統後或者discover工具執行後載入必須的驅動程式時隨然抽換您的網路硬體。

當核心偵測到新的硬體,則會初使硬體的驅動程式並執行 hotplug 程式來設定它。如果移除 硬體,則核心會再一次執行hotplug但以不同的環境變數。在 Debian 裡,當hotplug被呼叫 時,會執行/etc/hotplug/和/etc/hotplug.d/目錄底下的 script。更詳盡的訊息請參閱 hotplug(8)

新安裝的網路硬體是由/etc/hotplug/net.agent這個 script 來設定。<sup>13</sup> 假設您的 PCM-CIA 網路卡在安裝後使得 eth0 爲對應的網路裝置介面。則 /etc/hotplug/net.agent 會 完成底下的事情<sup>14</sup>:

ifup eth0=hotplug

除非您加入邏輯裝置介面定義或對應的hotplug名稱到/etc/network/interfaces,否則 該指定不會有任何效果。為了讓該指令能正確設定 eth0,加入下列敘述到/etc/network /interfaces:

mapping hotplug script echo

<sup>&</sup>lt;sup>13</sup>網路硬體也能被安裝在 /etc/hotplug.d/net/ 目錄下的 hook script 設定。例如,安裝ifplugd 和 waproamd套件會增加這些 hook script。

<sup>&</sup>lt;sup>14</sup>在 0.0.20040329-4 左右的版本, hotplug在某些情況下可以加入不同模式,使得它控制的方式跟往常不一樣。 有一種模式為 "all",hotplug會帶起所有的可熱抽換裝置介面。另外有一種模式為 "auto",hotplug只會帶起列 在/etc/network/interfaces中的 auto 敘述行的裝置介面。在這些模式裡,ifup 不會加入=hotplug附加參 數啓動。

如同'高階網路設定-ifupdow邏輯裝置定義'175 說明的,這將會對應上述的指令,也同等於 底下的指令:

ifup eth0=eth0

(如果您使用hotplug啓動時所帶起的ifplugd 或 waproamd事件來控制這些裝置介面,千萬 不要使用上述的對應敘述。)

如果您只希望eth0在熱抽換時被帶起,改用 grep 來取代 echo 敘述,如同下列:

mapping hotplug script grep map eth0

參閱 '神奇的網路設定方式' 176 和 /usr/share/doc/hotplug/README.Debian 取得更多 小技巧。

#### 10.8.3 觸發網路設定 – 使用ifplugd

ifplugd daemon 不管實體硬體有無安裝,都能帶起或關閉網路裝置介面。該程式能偵測乙太 網路是否接上纜線或是 Wi-Fi 裝置介面是否取得 AP(雖然waproamd是比較適合後來的情況)。 當 ifplugd 發現連線狀態改變時,它會執行一個代理 script,預設是呼叫 ifup 或 ifdown。

#### 10.8.4 觸發網路設定 - 使用 waproamd

waproamd daemon 類似ifplugd,只是說它是針對 Wi-Fi 介面卡設計的。它能動態掃描 Wi-Fi 硬體能存取的 AP。找到能存取的 AP 時, waproamd 會執行 ifup。

如果您使用 waproamd,则您應該直接使用waproamd來設定 Wi-Fi 介面卡而非使用/etc /network/interfaces中的wireless-\*敘述。

#### 10.8.5 網路設定和 PCMCIA

有許多可行的辦法來設定 PCMCIA 網路裝置介面(針對 2.4 和 2.6 的核心)。

- 針對 32 bit PCI (CardBus) PCMCIA 網路卡:
  - 由 hotplug 控制的 ifupdown
    - \* 在 Woody 或 Sarge 裡,您必須加入對應的敘述到/etc/network /interfaces使得hotplug能控制ifupdown,如同'觸發網路設定 – 使 用hotplug' 179的説明。
- 針對 16 bit ISA PCMCIA 網路卡:
  - 使用 hotplug 和 pcmcia-cs的限制來控制 ifupdown載入模組
    - \* Recommended

- \* 在 Woody 或 Sarge 裡,您必須加入exit 0這一行到/etc/pcmcia/network設定檔開頭來取消pcmcia-cs控制ifupdown的預設行為。同時,您必須加入對應的敘述到/etc/network/interfaces使得hotplug能控制ifupdown,如同'觸發網路設定-使用hotplug' 179的說明。
- pcmcia-cs控制ifupdown的方式是經由 /etc/pcmcia/network 的設定
   不贊成使用,但 Woody 和 Sarge 預設採用這個方式
- pcmcia-cs控制的低階的工具是經由 /etc/pcmcia/network 的特殊代碼所操縱 - 不贊成使用
  - 在 Woody 和 Sarge 裡, 編輯/etc/pcmcia/network.opts就能啓用特殊代碼

針對 16 bit 的介面卡而言,推薦使用核心 2.4,因爲該版本的 hotplug 子系統已經支援 PCMCIA 了。<sup>15</sup>

PCMCIA網路卡是可熱插拔的。因此,任何使用 PCMCIA 卡的網路服務必須要仔細設定,使得網路卡插入時能正常啓動;網路卡移除時能正常關閉服務。這通常是依據排列ifup和ifdown來啓動或關閉服務才能完成。有些人是限制他們的PCMCIA網路卡為不可 抽換:他們會在開機前安裝好網路卡,並在開機過程中陸續啓動需要該網路卡的服務。如果您 是這樣的朋友,為了確保該網路卡在啓動服務前是設定良好的。您必須完成下列事項:

- 在/etc/default/pcmcia設定CARDMGR\_OPTS="-f",這為了強迫cardmgr在背景執行。
- 更名 /etc/rc?.d/S20pcmcia 成類似 /etc/rc?.d/S12pcmcia。

這些動作只適合 16 bit PCMCIA 介面卡。

注意,如果您使用 16 bit PCMCIA 介面卡,還是需要安裝pcmcia-cs套件。該套件包含的cardmgr daemon 是用來管理 socket 和載入驅動程式模組。我們不用經由/etc/pcmcia/network去呼叫網路設定程式。

«««< gateway.sgml 為了讓cardmgr能正常工作,您可能要編輯/etc/pcmcia/config.opts來設定16 bit PCMCIA 介面卡的資源。更多訊息請參閱Linux PCM-CIA HOWTO (http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html) ======= 為了讓cardmgr能正常工作,您可能要編輯/etc/pcmcia/config.opts來設 定16 bit PCMCIA 介面卡的資源。更多訊息請參閱 Linux PCMCIA HOWTO (http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html) »»»> 1.16

# 10.9 多階段(Multi-stage)對應

第一個假設您的網路介面卡是可熱抽換的且您也設定好如'觸發網路設定 - 使用hotplug' 179 敘述的自動設定功能。第二個假設您需要對應邏輯裝置介面到"實體"裝置介面(如同'處理核心的非一致性裝置命名規則' 177的敘述)以及連接到網路的裝置介面(如同'使用 guessnet 選擇邏輯裝置介面' 177的敘述)。您可利用多階段對應來實現這個功能。

<sup>&</sup>lt;sup>15</sup>在先前發行的 Debian 版本裡,設定 PCMCIA 網路卡的標準動作是透過cardmgr的 hook script/etc/pcmcia/network 和 /etc/pcmcia/network.opts。這些hook scripts是比 Linux 獲得更多熱抽換機制前就已經開發 好了。 還是有些人使用 Debian Woody 的 scripts 為預設的解決方式,不管裝置介面加入或移除時,他們只 用ifup和ifdown來處理。現在推薦改使用 hotplug 來管理這些事情。 其餘的人仍然使用特別的啓動系統;當 /etc/pcmcia/network.opts中的特定變數被設定成 "y"時,直接呼叫低階的網路設定指令來啓動裝置介面。該 系統有些許的問題。在某些 race condition 下會出錯;它只能用在 16 bit 的 PCMCIA 卡; 留給 ifupdown 來處理 會比較恰當。所以不贊成使用該方式。

如果裝置介面可熱抽換,第一階段的對應則使用 hotplug 群組名並輸出核心所指定的裝置介面。第二階段對應使用核心指定的裝置介面名稱並輸出該介面卡的名稱。第三階段對應以網路 環境來對應介面卡名稱和邏輯裝置介面名稱。

```
# 允許 hotplug 帶起裝置
mapping hotplug
script echo
```

# 決定那一個介面是有線或無線

mapping eth?

```
script /usr/local/sbin/get-mac-address.sh
map 02:23:45:3C:45:3C wired
map 00:A3:03:63:26:93 wifi
```

# 偵測可取得的有線網路

```
mapping wired
    script guessnet-ifupdown
    map work-wired
    map home
```

```
# 偵測可取得的無線網路
```

```
mapping wifi
```

script ifscout map starbucks map work-wireless

```
iface work-wired inet static ...
```

# 10.10 網路服務設定

家用或桌上型電腦使用的典型網路服務包括:

- Internet super-server 和 TCP/IP daemon wrapper, 參閱'服務的存取限制' 123。
  - /etc/inetd.conf
- ssh: OpenSSH secure shell, 参閲 'SSH' 142.
  - /etc/ssh/ssh\_config
  - /etc/ssh/sshd\_config
- exim: 郵件傳送代理, 參閱 '郵件名稱' 168 和 '郵件傳送代理 (MTAs)' 146.
  - /etc/exim/exim.conf
  - /etc/mailname
  - /etc/aliases
  - /etc/email-addresses
- fetchmail: 從 POP3 帳號擷取郵件的 daemon, 參閱 '擷取信件1-Fetchmail' 149.
  - /etc/fetchmailrc

- procmail:本地端郵件傳送和分析器,參閱'處理郵件 Procmail' 149.
   ~/.procmailrc
- 主機名稱和 DNS (proxy, cache, ...), 參閱 '主機名稱' 168 和'網域名稱服務 (DNS)' 169.
  - /etc/host.conf
  - /etc/hostname
  - /etc/hosts
  - /etc/hosts.allow
  - /etc/hosts.deny
  - /etc/resolv.conf
  - /etc/bind/named.conf (編輯)
  - /etc/bind/db.lan (加入區網内的主機)
  - /etc/bind/db.192.168.0 (加入區網的反查對應)
- DHCP, 參閱 '安裝網路界面使用 DHCP' 171.
  - /etc/dhcp3/dhclient.conf (DHCP 客户端)
  - /etc/default/dhcp3-server(DHCP 伺服器端)
  - /etc/dhcp3/dhcpd.conf (DHCP 伺服器端)
- cvs: concurrent versions system, 参阅 'Concurrent Versions System (CVS)' 195.
  - /etc/cvs-cron.conf
  - /etc/cvs-pserver.conf
- nfs-kernel-server: network file system, 參閱'NFS設定' 35. (針對 unix-like 系統)
   /etc/exports
- samba: Windows 網路檔案和印表機分享, 參閱 'Samba設定' 35 and 'Samba' 117.
  - /etc/samba/smb.conf
- 印表機 daemon 系統, 參閱 '印表機設定' 36.
  - /etc/printcap(針對 lpr)
- apache 和 apache2: 網路伺服器.
  - /etc/apache/\*
  - /etc/apache2/\*
- squid: 網頁代理及快取伺服器.
  - /etc/squid/\*

# 10.11 解決網路問題

如果您在使用網路時出問題,請先執行下列的敘述來檢查輸出的結果:

```
# ifconfig
```

- # cat /proc/pci
- # cat /proc/interrupts
- # dmesg | more

同時,請參閱'網路測試基礎'113 接下來的段落。

如果您無法瀏覽特定的站台,請參閱'無法存取某此站台的怪問題'39。

# 10.12 建立通訊閘路由器

一台 Debian 主機能完成所有通訊閘機器的功能,諸如網路位址轉譯(NAT,也稱為 IP 偽裝),傳送郵件,DHCP,DNS 快取,HTTP 代理快取,CVS 服務,NFS 服務以及 Samba 服務。參考 (網路所需的主機名和IP地址'27 的範例來設定上述的服務。

#### 10.12.1 Netfilter 設定

netfilter/iptables 專案是核心 2.4 或之後的版本的一個防火牆子系統。參閱 Netfilter (http://www.netfilter.org/),您會取得許多網路設定的方針和說明。

#### Netfilter 基礎

Netfilter 使用五個內建的 "chain"-PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUT-ING 來處理封包。

rout	cing				
deci	ision				
PRE>	>	FORWARD	>>	POST>	OUT
ROUTING	$\setminus$	filter	/	ROUTING	interface
DNAT		tracking	^	SNAT	
REDIRECT			I	MASQUERADE	
	V		I		
II	NPUT		OUTPUT		
	filte	r	^ filte:	r,DNAT	
	V		I		
	\> Loo	cal Proce	ss/		
	use	er-space	programs		
	rout dec: PRE> ROUTING DNAT REDIRECT IN	routing decision PRE>> ROUTING \ DNAT   REDIRECT   v INPUT   filte: v \> Loo	routing decision PRE>> FORWARD ROUTING \ filter DNAT   tracking REDIRECT   v INPUT   filter v \> Local Proce user-space	routing decision PRE>> FORWARD> ROUTING \ filter / DNAT   tracking ^ REDIRECT     v   INPUT OUTPUT   filter ^ filter v   \> Local Process/ user-space programs	routing decision PRE>> FORWARD> POST> ROUTING \ filter / ROUTING DNAT   tracking ^ SNAT REDIRECT     MASQUERADE v   INPUT OUTPUT   filter ^ filter,DNAT v   \> Local Process/ user-space programs

#### Netfilter 表格

每一個內建的 "chain" 使用底下的表格來處理所有的封包。

- filter (預設使用的封包分析器)
  - INPUT (針對目標為本地主機的封包)
  - FORWARD (針對主機路由後的封包)
  - OUTPUT (針對本地主機產生的封包).
- nat (網路位址轉譯)
  - PREROUTING (針對進入主機所改變的封包)
  - OUTPUT (針對路由前所改變的本地主機產生的封包)
  - POSTROUTING (針對要送出所改變的封包)
- mangle (network address mangling, 只適用核心 2.4.18 之後的版本)
  - 以上是所有五個内建的鍊。

#### Netfilter 目標

防火牆的規則有幾種目標:

- 基本目標:
  - ACCEPT 表示讓封包通過。
  - DROP 表示捨棄封包。
  - QUEUE 表示傳遞封包到 userspace (如果核心支援的話)。
  - RETURN 表示停止讀取該鍊並回到前一個(呼叫的)鍊中的下一個規則。
- 擴充的目標:
  - LOG 啓動核心紀錄功能。
  - REJECT 回送一個錯誤封包並捨棄該封包。
  - SNAT 改變封包的來源位址,只用在 POSTROUTING 鍊。(只適用 nat 表格) --to-source ipaddr[-ipaddr][:port-port]
  - MASQUERADE 與 SNAT 相同但用在動態指定 IP (撥接)的連線。(只適用 nat 表格) --to-ports port[-port]
  - DNAT 改變封包的目的源位址,只用在 PREROUTING 和 OUTPUT 以及使用者組 合這兩種鍊的自訂鍊。(只適用 nat 表格)

--to-destination ipaddr[-ipaddr][:port-port]

- REDIRECT 改變封包的目的源位址並傳送封包到自己的機器。

--to-ports port[-port]

#### Netfilter 指令

```
基本的iptables指令爲:
                                     # 建立一個新 chain
  iptables -N chain
                                     # 加入規則到 chain
  iptables -A chain \setminus
           -t table \setminus
                                     # 使用 table (filter, nat, mangle)
           -p protocol \
                                     # tcp, udp, icmp, 或 all,
           -s source-address[/mask] \
                                    # 當 -p 是 tcp 或 udp 時,指定來
           --sport port[:port] \
源埠號
           -d destination-address[/mask] \
           --dport port[:port] \ # 當 -p 是 tcp 或 udp 時,指定目
的埠號
           -j target ∖
                                     # 如果符合則作何處理
                                    # 針對 INPUT, FORWARD, PREROUTING
           -i in-interface-name \
                                    # 針對 FORWARD, OUTPUT, POSTROUTING
           -o out-interface-name
```

#### 網路位址轉譯(NAT)

區網內的機器能夠存取網際網路的資源是因爲通訊閘轉譯了區網內的 IP 位置成網際網路合法的 IP 位置。

# apt-get install ipmasq

«««< gateway.sgml 使用範例的規則來加強ipmasq的保護。 ======= 使用範例的規則 來加強ipmasq的保護。參閱/usr/share/doc/ipmasq/examples/stronger/README。 »»»> 1.16

另外,如果是透過 PCMCIA 網路卡連線的,ipmasq 需要由 /etc/pcmcia /network.opts(閱讀 /usr/share/doc/ipmasq/ipmasq.txt.gz)或/etc/network /interfaces(閱讀'網路設定和 PCMCIA' 180 和'觸發(Triggering)網路設定' 178)來啓動。

#### 重導 SMTP 連線 (2.4)

假設您的筆記型電腦在不同的區網環境下,如果您不希望重新設定郵件用户代理程式就能直接 使用在筆記型電腦上。

執行 iptables 並加入底下的規則,它將會重導 SMTP 連線到通訊閘的機器。

至於完整的重導規則,安裝ipmasq套件並複製M30redirect.def(http://www.debian. org/doc/manuals/debian-reference/examples/)到/etc/ipmasq/rules/目 錄 下。

#### 10.12.2 管理多重的網路連線

[FIXME] Policy routing (by Phil Brutsche <pbrutsch@tux.creighton.edu>): 更多訊息請 參閱 iproute manual (http://lartc.org/)。Traffic control (tc) 也是有幫助的。

環境設定:

eth0: 192.168.1.2/24; 通訊閘 192.168.1.1 eth1: 10.0.0.2/24; 通訊閘 10.0.0.1 該機器沒有任何 IP 偽裝的機制。

Special magic:

- 1 ip rule add from 192.168.1.2 lookup 1
- 2 ip rule add from 10.0.0.2 lookup 2
- 3 ip route add to default via 10.0.0.1 metric 0
- 4 ip route add to default via 192.168.1.1 metric 1
- 5 ip route add table 1 to 192.168.1.0/24 via eth0
- 6 ip route add table 1 to 10.0.0.2/24 via eth1
- 7 ip route add table 1 to default via 192.168.1.1
- 8 ip route add table 2 to 192.168.1.0/24 via eth0

9 ip route add table 2 to 10.0.0.2/24 via eth1 10 ip route add table 2 to default via 10.0.0.2

[FIXME] 我從未試驗過。至於如何設定備緩的撥接以及自動重撥的功能來實現快速連線?還請 寄給我一個補丁:-)

# Chapter 11

# 编輯器

## 11.1 流行的编辑器

Linux 下有許多在控制台 (console) 環境下執行的文字編輯器任你選用,包括:

- vim: 強大而輕便的 BSD 傳統編輯器。VI iMproved。
- emacs: 重量級的 GNU 傳統編輯器。RMS (Richard M. Stallman) 原創。
- xemacs: Emacs 的下一代,由 Lucid 原創。
- mcedit: 初學者的 GNU 編輯器。也就是 mc 内建編輯器。參閱 'MC 的編輯器' 45。
- ae: 預設的小型編輯器 (Potato)。通常不用它。
- nano: 預設的小型 GNU 編輯器 (Woody)。模彷 pico。
- joe: 給舊式的 WordStar 或 TurboPascal。
- jed: 快速、多功能、選單式的編輯器,相容於 Emacs 的鍵盤操作方式。
- jove: 微型編輯器,相容於 Emacs 的鍵盤操作方式。
- nvi: 新版 vi。連 Bug 都相容於原本的 vi。

使用 update-alternatives --config editor 命令可改變預設的編輯器。許多程式也使用環境變數 EDITOR 或 VISUAL 來決定編輯器。參閱'MC 的編輯器'45。

還有一些於X環境下執行的編輯器也值得一提:

- gvim: Vim with GUI (vim 和 vim-gtk 套件)
- emacs: 唯一的 Emacs (自動偵測 X)。
- xemacs: 次世代 Emacs (自動偵測 X)。

這些 xclient 的命令列使用標準選項,如 -fn a24,這對像我這樣的老傢伙來說就再好不過了:)參閱 'X clients' 132。

# 11.2 應急的編輯器

有少數幾個編輯器安裝在 /bin 下。至少應該安裝一個這類的編輯器,以免 /usr/ 不能存取時,就無法編輯文件了。

- elvis-tiny: 最小的 vi 編輯器 (用 vi 命令啓動)
- nano: 最小的非 vi 編輯器 (用 nano-tiny 命令啓動)

• ed: 最小的編輯器 (總是存在但使用起來不方便)

# 11.3 Emacs 和 Vim

# 11.3.1 Vim 指示

程式執行時可按下 <F1> 閱讀 "VIM - main help file" 文件。

<f1></f1>	説明
<esc></esc>	返回正常模式
V	Visual 模式
i	Insert 模式
:	命令列指令
:set tw=72	設定文字行寬為 72
<f11></f11>	插入 (貼上) 模式
:r! date -R	插入 RFC-822 日期
qa	將鍵盤操作記錄到暫存器 a
q	停止鍵盤操作記錄
0 <i>a</i>	執行暫存器 a 的鍵盤操作
:edit <i>foo.txt</i>	載入並編輯檔案 foo.txt
:wnext	寫入目前的檔案並編輯下一個檔案

q和@可用來記錄簡單的鍵盤巨集然後執行它們。例如,想建立一個巨集爲將游標所在處的 單詞加上HTML 斜體字標籤,可以輸入qii<i>> [ea</i>^ [q(此處 ~ [表示按ESC 鍵)。然後,在單詞前輸入@i,編輯器就會自動爲它加上 <i>和 </i>>

也請參閱'在 Vim 中使用 GnuPG' 225。

### 11.3.2 Emacs 提示

<f1></f1>	説明	
<f10></f10>	選單	
C-u M-! date -R	插入 RFC-822 日	期

### 11.3.3 啓動編輯器

啓動編輯器:	emacs filename	vim filename
以與 vi 相容方式啓動:		vim -C
以與 vi 不相容方式啓動:		vim -N
以編譯預設值啓動:	emacs -q	vim -N -u NONE

# 11.3.4 Editor 指令摘要 (Emacs , Vim)

exit:	С-х С-с	:qa /:wq /:xa /:q!
Get back/command mode:	C-g	<esc></esc>
Backward(left):	C-b	h
Forward(right):	C-f	1
Next(down):	C-n	j
Previous(up):	С-р	k
<pre>stArt of line(^):</pre>	C-a	0
End of line(\$):	C-e	Ş
mUltiple commands:	C-u nnn cmd	nnn cmd
Multiple commands:	M-digitkey cmd	
save File:	C-x C-s	:w file
beginning of buffer:	M-<	1G
end of buffer:	M->	G
scroll forward 1 screen:	C-v	^F
scroll forward 1/2 screen:		^D
scroll forward 1 line:		^E
scroll backward 1 screen:	M-v	^B
scroll backward 1/2 screen:		^U
scroll backward 1 line:		^Υ
scroll the other window:	M-C-v	
delete under cursor:	C-d	х
delete from cursor to eol:	C-k	D
iSearch forward:	C-s	
isearch Reverse:	C-r	
Search forward:	C-s enter	/
search Reverse:	C-r enter	?
isearch regexp:	M-C-s	
isearch backward regexp:	M-C-r	
search regexp:	M-C-s enter	/
search backward regexp:	M-C-r enter	?
Help:	C-h C-h	:help
Help Apropos:	C-h a	
Help key Bindings:	C-h b	:help [key]
Help Info:	C-h i	
Help Major mode:	C-h m	
Help tutorial:	C-h t	:help howto
Undo:	C	u
Redo:	C-f	^R
Mark cursor position:	С-б	$m\{a-zA-Z\}$
eXchange Mark and position:	C-x C-x	
goto mark in current file:		′{a−z}
goto mark in any file:		′ { A-Z }
copy region:	M-w	{visual}y
kill region:	C-w	{visual}d

Yank and keep buffer: Yank from kill buffer: convert region to Upper: convert region to Lower: Insert special char: replace: replace regexp: query replace: query replace: query replace reqexp: Open file: Save file: Save all buffers: Save as: Prompt for buffer: List buffers: Toggle read-only: Prompt and kill buffer: Split vertical: Split horizontal: Move to other window: Delete this window: Delete other window(s): run shell in bq: kill shell run in bg: run make: check error message: run shell and record: ....clean BS, ... ... save/recall shell record: run shell: run command: run command and insert: run filter: run filter and insert: show option reset option to default reset boolean option toggle boolean option wrap text at column 72 do not wrap autoindent expand tab specify comment (mail)

C-y M-y р C-x C-u {visual}U C-x C-l {visual}u C-q octalnum/keystroke ^V decimal/keystroke M-x replace-string :%s/aaa/bbb/q M-x replace-regexp :%s/aaa/bbb/q M−% :%s/aaa/bbb/gc M-x query-replace M-x query-replace-regexp :r file C-x C-f C-x C-s :W C-x s :wa C-x C-w file :w file C-x b C-x C-b :buffers C-x C-q :set ro C-x k C-x 2 :split C-x 3 :vsplit (ver. 6) C-x o ^Wp C-x 0 :q ^Wo C-x 1 M-x compile M-x kill-compilation :make Makefile С-х ' :echo errmsq :!script -a tmp M-x shell :!col -b <tmp >record C-x C-w record :r record M-! sh :sh M-! cmd :!cmd C-u M-! cmd :r!cmd M-| file {visual}:w file C-u M-| filter {visual}:!filter :se[t] {option}? :se[t] {option}& :se[t] no{option} :se[t] inv{option} :se tw=72 :se tw=0 :se ai :se et :se comments=n:>, n:\|

run GDB	M-x gdb
describe GDB mode	C-h m
step one line	M-s
next line	M-n
step one instruction (stepi)	M-i
finish current stack frame	C-c C-f
continue	M-c
up arg frames	M-u
down arg frames	M-d
copy number from point, insert	at the end
	C-x &
set break point	C-x SPC

### 11.3.5 Vim 設定

要使用 Vim 所有的功能和高亮度關鍵字,請在 ~/.vimrc 或 /etc/vimrc 加入下列内容:

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Paste 模式能避免自動縮進功能影響在控制台終端下的剪下一貼上操作,比只用 ":set noai"更好。

有關 GnuPG 整合的資訊,請參閱 '在 Vim 中使用 GnuPG' 225。

#### 11.3.6 Ctags

執行 apt-get install exuberant-ctags,接著就可以在源碼檔中使用 ctags 了。在 Vim 中輸入:tag function\_name 可直接跳到 function\_name 的開始行。它適用於 C、 C++ 、 Java 、 Python 和許多其它的語言。

Emacs 也有相同的 ctags 功能。

#### 11.3.7 將高亮度顯示的螢幕内容轉為 HTML 文件

在 Vim 命令模式下輸入 so \\$VIMRUNTIME/syntax/2html.vim,可將螢幕上高亮度顯示的文字轉化為 HTML 文字,然後用:w file.html 存檔, :q 退出。對 C 等源碼特別適用。

#### 11.3.8 用 vim 分割螢幕

vim 可以在多重分割視窗 (multi-split-screen) 環境下編輯多個檔案。想了解有關詳情,可輸入 :help usr\_08.txt。

要分割螢幕顯示多個不同文件,請在 vi 命令提示後輸入:

:split another-file
:vsplit another-file

或在 shell 提示時:

\$ vi -o file1.txt file2.txt # 水平分割
\$ vi -O file1.txt file2.txt # 垂直分割

就可以使用多視窗 vi。

\$ vimdiff file.txt~ file.txt # 檢查 file.txt 的最近修改情况
\$ vimdiff file.en.sgml file.fr.sgml # 檢查翻譯的變化
\$ gvimdiff file.txt~ file.txt # 在 X 中

上述操作可以明確地顯示原始文件與備份文件的差別。對 SGML 文件,它進行標籤匹配檢查, 所以用它來檢查翻譯結果十分有效。

用 CTRL-W 命令指定游標移動:

Ctrl-W	+	增加視窗大小
		试小祖宫十小
Ctrl-W	_	成小沉因八小
Ctrl-W	h	移動主視窗左邊
Ctrl-W	j	移動至視窗下方
Ctrl-W	k	移動至視窗上方
Ctrl-W	1	移動至視窗右邊

下列命令用於控制螢幕捲軸:

:set scrollbind
:set noscrollbind

# Chapter 12

# 版本控制系統

# 12.1 Concurrent Versions System (CVS)

有關的詳細訊息可使用 lynx 查閱 /usr/share/doc/cvs/html-cvsclient , /usr/share/doc/cvs/html-info, /usr/share/doc/cvsbook 或執行 info cvs 及 man cvs。

### 12.1.1 安裝 CVS 伺服器

以下設定的步驟,僅允許 "src" 群組的成員存取 CVS 檔案庫,並且僅 "staff" 群組的成員才可 管理 CVS,這樣做可以降低管理者不小心犯錯的機會。

```
# cd /var/lib; umask 002; mkdir cvs # [Woody] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src . # "staff" to restrict more for starting project.
# chmod 3775 . # 如果上面使用 "staff",則這裡用 2775
# cvs -d /var/lib/cvs init # 在此明確指定 -d 會更安全!
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

### 12.1.2 CVS session 範例

下面是設定 shell 環境以便存取 CVS 檔案庫 (repository)。

#### 匿名 CVS (僅能下載)

遠端唯讀存取:

- \$ export CVSROOT=:pserver:anonymous@cvs.sf.net:/cvsroot/qref
  \$ cvs login
  \$ cvs -z3 co qref

使用本地 CVS 伺服器

\$ export CVSROOT=/var/lib/cvs

透過同一台機器上的 shell 進行本地端存取:

使用遠端 CVS pserver

不使用 SSH (在 cvs 中使用 RSH 協議) 遠程存取:

\$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
\$ cvs login

這個方法有被竊聽攻擊的疑慮。

#### 透過 ssh 使用遠端 CVS

透過 SSH 進行遠端存取:

\$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs

#### 或以 SourceForge 爲例:

\$ export CVSROOT=:ext:account@cvs.sf.net:/cvsroot/qref

你亦可使用 RSA 認證 ('以少許密碼連線 - RSA' 144),它不需要密碼提示。

#### 建立新的 CVS 庫存 (archive)

#### 想要建立如下的檔案,

ITEM	VALUE	MEANING
source tree:	~/project-x	所有的源碼
Project name:	project-x	專案名稱
Vendor Tag:	<i>Main-branch</i>	整個分支的標記 (Tag)
Release Tag:	Release-initial	特定分支的標記

则,

\$ cd ~/project-x	#	進源碼目錄		
建立源碼樹				
\$ cvs import -m "Start project-x	"	project-x	Main-branch	Release-initial
\$ cd; rm -R ~/project-x				

#### 使用 CVS

使用本地端 CVS 檔案庫來進行 project-x:

```
# 轉到工作區
$ cd
                            # 從 CVS 下載源碼到本地端
$ cvs co project-x
$ cd project-x
... 修改内容 ...
> cvs allI -u# 相當於 diff -u$ cvs up -C modified_file# 取消對文件的修改
$ cvs diff -u
                           # 相當於 diff -u repository/ local/
$ cvs ci -m "Describe change" # 將本地端源碼存到 CVS 上
$ vi newfile_added
$ cvs add newfile_added
$ cvs ci -m "Added newfile_added"
                            # 從 CVS 合併最新版本
$ cvs up
... 要從 CVS 建立所有新建立的子目錄的話,請改用 "cvs up -d -P"
 ... 注意以 "C filename" 作開頭的行
... 沒有變動的碼會移至 `.#filename.version'
 ... 在 filename 中搜尋 "<<<<<" 和 ">>>>>>"
$ cvs tag Release-1
                            # 添加 release tag
... 繼續編輯 ...
$ cvs tag -d Release-1 # 移除 release tag
$ cvs ci -m "more comments"
                            # 重新加入 release tag
$ cvs tag Release-1
$ cd
                            # 返回工作區
$ cvs co -r Release-initial -d old project-x
 ... 取得原版本並放入 old 目錄
```

```
$ cd old
  $ cvs tag -b Release-initial-bugfixes # 建立 branch (-b) tag
   ... 現在你可以對舊版本做修改了 (Tag=sticky)
  $ cvs update -d -P
   ... 源碼樹現在有 "Release-initial-bugfixes" 的 sticky tag
   ... 對分支做修改
  $ cvs up -d -P # 將在這個分支版本上被他人修改的檔案同步化
  $ cvs ci -m "check into this branch"
  $ cvs update -kk -A -d -P
   ... 移除 sticky tag 並取消内容
   ... update from main trunk without keyword expansion
  $ cvs update -kk -d -P -j Release-initial-bugfixes
   ... 將 Release-initial-bugfixes 的分支合併至主要版本
   ... trunk without keyword expansion. Fix conflicts with editor.
  $ cvs ci -m "merge Release-initial-bugfixes"
  $ cd
  $ tar -cvzf old-project-x.tar.gz old #建立壓縮備份, -j可使
用 bz2 格式
                               # 移除本地端源碼 (可選)
  $ cvs release -d old
```

應該記住的幾個選項(用作 cvs 命令行的第一個參數):

-n dry run, no effect-t display messages showing steps of cvs activity

#### 從 CVS 匯出 (export) 檔案

要從 CVS 獲得最新版本,用"tomorrow":

\$ cvs ex -D tomorrow module\_name

#### 管理 CVS

爲專案加上別名 (alias) (本地伺服器):

```
$ su - admin  # staff 成員之一
$ export CVSROOT=/var/lib/cvs
$ cvs co CVSROOT/modules
$ cd CVSROOT
$ echo "px -a project-x" >>modules
$ cvs ci -m "Now px is an alias for project-x"
$ cvs release -d .
$ exit  # 或 Ctrl-D 自 su 返回
```
\$ cvs co -d project px
... 自 CVS 中匯出 project-x (別名: px) 至 direcotry 專案
\$ cd project
... 修改内容 ...

#### 12.1.3 CVS 常見問題解決方法

#### 容器庫中的檔案權限

CVS 不會覆蓋當前容器中的文件,而是用另一個文件替換它。因此,對檔案庫目錄的寫入權 限是很重要的權限。所以如果需要的話,在新建檔案庫時請執行下面的命令,以確保權限的設 定。

# cd /var/lib/cvs
# chown -R root:src repository
# chmod -R ug+rwX repository
# chmod 2775 repository # 如果需要的話,對子目錄也做同樣的設定

#### 可執行權限 (Execution bit)

當文件被別人匯出後會保留可執行權限。任何時候當你匯出的文件遇到執行權限的問題,可用 下面的命令在 CVS 檔案庫中修改文件的權限。

# chmod ugo-x filename

# 12.1.4 CVS 命令

這裡是一些 CVS 命令和他們的捷徑的用法。

```
{add|ad|new} [-k kflag] [-m 'message'] files...
{admin|adm|rcs} [rcs-options] files...]
{annotate|ann} [options] [files...]
{checkout|co|get} [options] modules...
{commit|ci|com} [-lnR] [-m 'log_message' | -f file] \
        [-r revision] [files...]
{diff|di|dif} [-k1] [rcsdiff_options] [[-r rev1 | -D date1] \
        [-r rev2 | -D date2]] [files...]
{export|ex|exp} [-flNn] -r rev|-D date [-d dir] [-k kflag] module...
{history|hi|his} [-report] [-flags] [-options args] [files...]
{import|im|imp} [-options] repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-1] rlog-options [files...]
```

# 12.2 Subversion

Subversion 是下一代版本控制系統,它將替代 CVS。當前開發者稱它還處於 "alpha" 階段,但 對大多數用戶而言它已足夠穩定了。到本文檔寫作之時,Subversion 僅在 Debian unstable 中可用。

#### 12.2.1 安裝 Subversion 伺服器

subversion meta-package 依 賴 一 些 相 依 的 套 件 (libapache2-svn 和 subversion-tools) 來設定伺服器。

## 設立檔案庫

當前, subversion 無法建立檔案庫,所以使用者需要手動創建它們。通常會在/var/local/repos 下建立檔案庫。

#### 創建目錄:

# mkdir -p /var/local/repos

建立檔案資料庫:

# svnadmin create /var/local/repos

#### 將容器的寫權限賦給 www server:

# chown -R www-data:www-data /var/local/repos

#### 設定 Apache2

想要用使用者認證來授權存取檔案庫,則添加(或去除注釋)下列内容到 /etc/apache2 /mods-available/dav\_svn.conf:

```
<Location /repos>
DAV svn
SVNPath /var/local/repos
AuthType Basic
AuthName "Subversion repository"
AuthUserFile /etc/subversion/passwd
<LimitExcept GET PROPFIND OPTIONS REPORT>
Require valid-user
</LimitExcept>
</Location>
```

接著,使用下面的命令建立使用者認證檔:

htpasswd2 -c /etc/subversion/passwd some-username

重新啓動 Apache2,就可以使用 URL http://hostname/repos 來存取新的 subversion 檔案庫 了。

## 12.2.2 將 CVS 檔案庫轉移到 Subversion

#### 12.2.3 Subversion 使用範例

下面的小節將教你如何在 Subversion 下使用各種命令。

#### 建立新的 Subversion 保存檔(archive)

要建立新的 Subversion 保存檔,請輸入下面的命令:

\$ cd ~/your-project # 進入你的源碼目錄

\$ svn import http://localhost/repos your-project \
 project-name -m "initial project import"

這將在你的 Subversion 檔案庫下建立一個名爲 project-name 的目錄,用來存放你的專案文件。 查看 http://localhost/repos/ 它是否在那兒?

#### 使用 Subversion

```
用 Subversion 來管理 project-y:
                                # 進入工作區域
  $ cd
  $ svn co http://localhost/repos/project-y # 匯出源碼
  $ cd project-y
   ... 完成一些工作 ...
  $ svn diff
                               # 相當於 diff -u repository/ local/
  $ svn revert modified file
                               # 取消對文件所做的修改
  $ svn ci -m "Describe changes" # 將你做的修改匯入至檔案庫中
  $ vi newfile added
  $ svn add newfile_added
                               # 將 new_dir 中的所有文件遞迴地加入檔
  $ svn add new dir
案庫中
                               # 將 new_dir2 中的文件非遞迴地加入檔案
  $ svn add -N new_dir2
庫中
  $ svn ci -m "Added newfile_added, new_dir, new_dir2"
                               # 從檔案庫中合併最新的版本
  $ svn up
                                # 顯示所有已提交的修改記錄
  $ svn log
  $ svn copy http://localhost/repos/project-y \
        http://localhost/repos/project-y-branch \
        -m "creating my branch of project-y" # 建立 project-y 的分支
版本
  $ svn copy http://localhost/repos/project-y \
        http://localhost/repos/proj-y_release1.0 \
        -m "project-y 1.0 release" # 加入 release tag
   ... 注意,分支 (branching) 和標記 (tagging) 是相同的。唯一的不同在分支會
提交 (committed) 而標記不會。
   ... 對分支版本做修改 ...
  s # 將分支版本回存至主要版本
  $ svn merge http://localhost/repos/project-y \
```

http://localhost/repos/project-y-branch

```
$ svn co -r 4 http://localhost/repos/project-y # 取得第四版本
```

# Chapter 13

# 程式設計

不要把測試用的執行檔名稱取成"test"。 test 是一個 shell 的内建指令。

# 13.1 從哪兒開始

參考資料:

- 在 /usr/share/doc/package 中的文件和範例
- Unix / Programming Information (http://arioch.unomaha.edu/~jclark/ #info)
- *Linux Programming Bible* (John Goerzen/IDG books)

從 GNU (http://www.gnu.org/) 可以獲得更多可印成紙本的豐富文件。

接下來的四個小節包含了用不同程式語言所寫的簡單 script ,其功能是建立一個文字檔,其內容可提供給像 newusers 之類的批次執行程式來使用,其功能是用來加入 /etc/passwd 的帳號資料。每個 script 都需要一個輸入檔,其每行都是像 first\_name last\_name password 這樣的格式。(這些 script 會建立實際的使用者 home 目錄。)

# 13.2 Shell

了解 Unix-like 系統如何工作的最好方法就是閱讀 shell script。在此,我們對 shell 编程做個 簡單的重點提示。從錯誤中學習,請參閱 Shell Mistakes (http://www.greenend.org.uk/ rjk/2001/04/shell.html)。

### 13.2.1 Bash-GNU標準互動式 shell

Bash 的参考資料:

- bash(1)
- info bash

- the LDP BASH Programming Introduction HOWTO (http://www.tldp.org/ HOWTO/Bash-Prog-Intro-HOWTO.html) 是初學者的資訊來源。
- mc /usr/share/doc/bash/examples/ /usr/share/doc/bash/ (安 装 bash-doc 套件可以看到更多的範例檔。)
- *Learning the bash Shell*, 2nd edition (O'Reilly)

簡短的程式範例(從標準輸入讀取輸入,可提供 newusers 來建立帳號資訊):

```
#!/bin/bash
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

# 13.2.2 POSIX shells

Debian 中有幾個套件提供了 POSIX shell:

- dash (Sarge)
  - Priority: optional
  - Installed-Size: 176
  - 目前爲止最小而且最快的-用來開機初始的最佳選擇
- ash (Woody)
  - Priority: optional
  - Installed-Size: 180
  - 較小並且較快的 用來開機初始的好選擇
- bash
  - Essential: yes
  - Priority: required
  - Installed-Size: 580
  - 較大並且功能完整 實作了很多擴充功能
- pdksh
  - Priority: optional
  - Installed-Size: 408
  - 完整 AT&T ksh 的外觀和風格

如果你想編寫具有可攜性的 shell script ,最好能寫成 POSIX shell script。可將 /bin/sh 鏈接 到 ash (或 dash) 來測試和 POSIX 的相容性。最好能避免寫出包含 "bashisms" 或 "zshisms" 的 script 。例如,應避免下面的用法:

- if [ foo == bar ] ; then ...
- diff -u *file*.c{.orig,}
- mkdir /foo{bar,baz}

在這份文件中的敘述只適用於 POSIX 的 shell ,而不適用於 csh 式的 shell ,包括 tcsh 。

#### 13.2.3 Shell 參數

有幾個特別的參數是應該要記得的:

\$0	= shell 的名稱或 shell script 的名稱
\$1	= 第一個 shell 參數
• • •	
\$9	= 第九個 shell 參數
\$#	= 參數的個數
"\$*"	= "\$1 \$2 \$3 \$4 \$ <i>n</i> "
"\$@"	= "\$1" "\$2" "\$3" "\$4" "\$ <i>n</i> "
\$?	= 最近執行命令的退出狀態
\$\$	= 目前 shell script 的 PID
\$!	= 最近啓動的背景執行工作的 PID

要記住的基本延伸參數:

```
從  如果 var 設定的話  如果 var 沒有設定的話

${var:-string} $var string

${var:+string} string null

${var:=string} $var string

$var string
```

在這裡,這些操作中的冒號??都是選擇性的。

- 有 ':' = 操作中测試 "存在" 和 "非空"。
- 沒有 '!' = 操作中只測試 "存在"。

需要記住的基本參數代換:

從	結果		
\${var%suffix}	移除最小符合	suffix	的後綴形式
\${var%%suffix}	移除最大符合	suffix	的後綴形式
\${var#prefix}	移除最小符合	prefix	的前缀形式
\${var##prefix}	移除最大符合	prefix	的前缀形式

#### 13.2.4 Shell 重導 (redirection)

需要記住的基本重導 (redirection)用法 (在此 [n] 是可指定的數字,代表檔案描述子 (file de-scriptor)):

```
[n]> file 將 stdout (或 n) 重導至 file。
[n]>> file 將 stdout (或 n) 附加至 file。
[n]< file 由 file 重導至 stdin (或 n)。</li>
[n1]>&n2 將 stdout (或 n1) 重導至 n2。
2> file>&2 將 stdout 和 stderr 重導至 file。
> file 2>&1 將 stdout 和 stderr 重導至 file。
| command 將 stdout 和 stderr 通過 pipe 傳遞給 command。
```

在這裡,

- stdin: 標準輸入 (檔案描述子=0)
- stdout: 標準輸出 (檔案描述子=1)
- stderr: 標準錯誤 (檔案描述子 = 2)

shell 允許你用其内建的 exec,以任意的檔案描述子來開啓檔案。

```
$ echo Hello >foo
$ exec 3<foo 4>bar # open files
$ cat <&3 >&4 # redirect stdin to 3, stdout to 4
$ exec 3<&- 4>&- # close files
$ cat bar
Hello
```

在這裡的 n<&- 和 n>&- 是關閉檔案描述子 n。

## 13.2.5 Shell 條件式

每個命令均可回傳一個退出狀態,而這個狀態值可用於條件判斷式:

- 成功:0(真值)
- 錯誤: 1-255 (偽値)

注意此處的用法,回傳值0用來表示"眞值,與計算機其它領域中常見的表示是不同的。另外 们等同於使用 test 命令,並會將與17間的文字當成參數,所以相當於一個條件判斷式。

需要記住的基本條件式慣用法:

command && "如果前指令成功的話,也要執行這個指令" || true command || "如果前指令失敗的話,就執行這個指令"

if [ 條件判斷式 ]; then "如果前指令成功的話,也要執行這個指令" else "如果前指令失敗的話,就執行這個指令"

fi

在這裡的 || true 是需要的,以確保如果是用 -e 的選項在執行這行時,不會意外地結束程式。

在條件判斷式中的檔案比較有:

-e <i>file</i>	如果」	file 存在	則爲真。				
-d file	如果	file 存在	且為一目	錄則為	影。		
-f file	如果」	file 存在	且為一般	檔案則	爲眞。		
-w file	如果」	file 存在	且可寫入	則為真	0		
-x file	如果	file 存在	且可執行	則爲眞	0		
file1 -nt file	e2 如果 ]	file1 比	file2	新的話貝	刂爲真。	(modification	)
file1 -ot file	e2 如果 ]	file1 比	file2	舊的話貝	刂爲真。	(modification	)
file1 -ef file	2 如果雨	者的 dev:	ice 和	inode	號碼都相	同則爲真。	

在條件判斷式中的字串比較有:

-z str 如果 str 的長度是零則為真。
-n str 如果 str 的長度非零則為真。
str1 == str2 如果字串相等則為真。
str1 = str2 如果字串相等則為真。
(為了嚴格遵守和 POSIX 相容性,應用 "==" 來取代 "=")
str1 != str2 如果字串不相等則為真。
str1 < str2 如果 str1 的排列順序在 str2 之前則為真 (和 locale 相關)。</li>
str1 > str2 如果 str1 的排列順序在 str2 之後則為真 (和 locale 相關)。

在條件判斷式中的整數算術比較有 -eq、 -ne、 -lt、 -le、 -gt和 -ge。

#### 13.2.6 命令列操作

Shell 會用下列的方法來處理 script :

- 以中介字元來切分 tokens,有: SPACE、TAB、NEWLINE、;、(、)、<、>、
   1、&
- 如果不在"..." 或'...' 内的話, 會檢查關鍵字 (loop)
- 如果不在 "..." 或 '...' 内的話, 會展開alias (別名) (loop)
- 如果不在"..." 或'...' 内的話,會展開大括號, a {1,2} -> a1 a2
- 如果不在 "..." 或 '...' 内的話, 會展開tilde, ~user -> user 的家目錄
- 如果不在'...'内的話,會展開參數, \$PARAMETER
- 如果不在'...'内的話,會展開命令取代式, \$(command)
- 如果不在"..." 或'...' 内的話, 會以 \$IFS 來切分文字
- 如果不在"..." 或 '...' 内的話,會展開路徑名稱中的 \*?[]
- 搜尋命令
  - 函式
  - 内建
  - \$PATH 中的檔案
- loop

在單引號中的雙引號是沒有作用的。

Executing set -x in the shell or invoking the shell with -x option make the shell to print all of commands executed. This is quite handy for debugging.

### 13.3 Awk

Awk 的參考資料:

- Effective awk Programming, 3rd edition (O'Reilly)
- *Sed & awk,* 2nd edition (O'Reilly)
- mawk(1) 和 gawk(1)

• info gawk

簡短的程式範例(從標準輸入讀取輸入,可提供 newusers 來建立帳號資訊):

```
#!/usr/bin/awk -f
# Script to create a file suitable for use in the 'newusers' command,
# from a file consisting of user IDs and passwords in the form:
# first_name last_name password
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Distributed under GNU GPL v 2, or at your option, any later version.
# This program is distributed WITHOUT ANY WARRANTY.
BEGIN {
    # Assign starting UID, GID
    if ( ARGC > 2 ) {
        startuid = ARGV[1]
        delete ARGV[1]
    }
    else {
        printf( "Usage: newusers startUID file\n" \
          "
             where:\n" \
          "
               startUID is the starting userid to add, and \ \
          "
               file is an input file in form:\n" \
          "
                 first_name last_name password\n" \
        )
        exit
    }
    infile = ARGV[1]
    printf( "Starting UID: %s\n\n", startuid )
}
/^#/ { next }
{
    ++record
    first = $1
    last = $2
    passwd = $3
    user= substr( tolower( first ), 1, 1 ) tolower( last )
    uid = startuid + record - 1
    gid = uid
    printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n", \
        user, passwd, uid, gid, first, last, user \
        )
}
```

- 在 Debian 中提供 POSIX awk 的套件:
  - mawk
    - Priority: required
    - Installed-Size: 228
    - 較小並且快得多 適合於預設安裝
    - Compile-time limits exist
      - \* NF = 32767
        - \* sprintf buffer = 1020
  - gawk
    - Priority: optional
    - Installed-Size: 1708
    - 較大並且功能多 -- 實作很多延伸功能
      - \* System V Release 4 version of UNIX
      - \* Bell Labs awk
      - \* GNU-specific

# 13.4 Perl

這是 Unix-like 環境中最重要的直譯器。

Perl 的參考資料:

- perl(1)
- Programming Perl, 3rd edition (O'Reilly)
- The Perl Directory (http://www.perl.org/)

簡短的程式範例(從標準輸入讀取輸入,可提供 newusers 來建立帳號資訊):

安裝 Perl 模組 (model) module\_name:

# perl -MCPAN -e 'install module\_name'

# 13.5 Python

這是一個不錯的物件導向直譯器

Python 的參考資料:

- python(1)
- *Learning Python* (O'Reilly).
- Python Programming Language (http://www.python.org/)

簡短的程式範例(從標準輸入讀取輸入,可提供 newusers 來建立帳號資訊):

```
#! /usr/bin/env python
import sys, string
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Ported from awk script by KMSelf Sat Aug 25 20:47:38 PDT 2001
# This program is distributed WITHOUT ANY WARRANTY.
def usages():
   print \
"Usage: ", sys.argv[0], " start_UID [filename]\n" \
"\tstartUID is the starting userid to add.\n" \
"\tfilename is input filename. If not specified, standard input.n\n" \
"Input file format:\n"\
"\tfirst name last name password\n"
                return 1
def parsefile(startuid):
   #
    # main filtering
    #
   uid = startuid
   while 1:
        line = infile.readline()
        if not line:
           break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # above crashes with wrong # of parameters :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" % \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid
```

```
if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
        #print "# UID start from: %d\n" % uid
        if len(sys.argv) > 1:
            infilename = string.join(sys.argv[2:])
            infile = open(infilename, 'r')
            #print "# Read file from: %s\n\n" % infilename
        else:
            infile = sys.stdin
        parsefile(uid)
```

# 13.6 Make

Make 的參考資料:

- info make
- make(1)
- *Managing Projects with make*, 2nd edition (O'Reilly)

簡單自動變數:

規則 (rule) 的語法:

target: [ prerequisites ... ] [TAB] command1 [TAB] -command2 # 忽略錯誤 [TAB] @command3 # 抑制 echo

這裡的 [TAB] 就是一個 TAB。在 make 完成變數代換後,將會用 shell 逐行執行。在行尾使用 \ 可以續行。使用 \$\$ 可在 shell script 的環境中輸入 \$ 。

適用於 target 和 prerequisites 的隱含規則 (Implicit rules),如下的例子:

%: %.c header.h

#### 或,

%.o: %.c header.h

在此, target 包含了 % (exactly one of them), % 可匹配實際的目標檔名中任何的非空子串。 prerequisites 同樣也使用 % 來顯示它們的名稱是如何關聯到實際的目標檔名的。

後綴規則 (Suffix rules) 是用來定義 make 隱含規則的過時方法。GNU make 為了相容性的考量仍支持它,但只要可能的話就應該使用與之等價的規則:

舊的後綴規則	> 新的規則
.C:	> % : %.C
.c.o:	> %.0: %.C

規則中的自動變數:

foo.o: new1.c new2.c old1.c new3.c \$@ == foo.o (目標) \$< == new1.c (第一個) \$? == new1.c new2.c new3.c (有更新的) \$^ == new1.c new2.c old1.c new3.c (所有的) \$\* == 在 target 中、%'所匹配的文字。

變數參考:

```
fool := bar # 舊時的 expansion
foo2 = bar # 遮迴的 expansion
foo3 += bar # 附加
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS = $(dir directory/filename.ext) # 取出 "directory"
$(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

執行 make -p -f/dev/null 可查看内部自動規則。

# 13.7 C

#### 準備工作:

# apt-get install glibc-doc manpages-dev libc6-dev gcc

#### C 的參考資料:

- info libc (C 函式庫函式參考資料)
- gcc(1)
- each\_C\_library\_function\_name(3)
- Kernighan & Ritchie, *The C Programming Language*, 2nd edition (Prentice Hall). http://cm.bell-labs.com/cm/cs/cbook/
- ISO C standard. http://www.open-std.org/jtc1/sc22/wg14/

### 13.7.1 簡單的 C 程式 (gcc)

```
一個簡單的範例,將 example.c 和 libm 函式庫一起編譯,產生執行檔 run example:
   $ cat > example.c << EOF</pre>
   #include <stdio.h>
   #include <math.h>
   #include <string.h>
   int main(int argc, char **argv, char **envp) {
           double x;
           char y[11];
           x=sqrt(argc+7.5);
           strncpy(y, argv[0], 10); /* prevent buffer overflow */
           y[10] = ' \setminus 0'; /* fill to make sure string ends with ' \setminus 0' * /
           printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
           return 0;
   }
   EOF
   $ gcc -Wall -g -o run_example example.c -lm
   $ ./run_example
           1, 2.915, ./run_exam,
                                       (null)
   $ ./run_example 1234567890qwerty
           2, 3.082, ./run_exam, 1234567890qwerty
```

在這裡,因爲用到了 sqrt(),所以必需要用-1m 來連結 libm 函式庫。實際上的函式庫是位於/lib/中的 libm.so.6,而這是個符號連結至 libm-2.1.3.so。 請注意輸出文字的最後一個參數。即使已經指定了%10s,仍會超過 10 個字元。 不建議使用沒有邊界檢查的記憶體指標操作,如 sprintf 和 strcpy。爲預防暫存溢位的問題,請以 snprintf 和 strncpy 來取代。

#### 13.7.2 除錯

#### 用 gdb 來除錯

準備工作:

# apt-get install gdb

gdb 參考資料:

- info gdb (tutorial)
- gdb(1)
- http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html

使用 gdb 的 -g 選項可以用來除錯已編譯的程式。很多命令都可以縮寫。 Tab expansion 的功能和在 shell 時一樣。

在 Emacs 下做除錯的工作,請參閱 'Editor 指令摘要 (Emacs, Vim)' 191。

由於編譯好的軟體會被拆解成數個套件安裝到 Debian 系統,大部分的 debugging symbols 是 被移除的。為了讓 gdb 能有效地除錯 Debian 套件,相關的套件在重新編譯時需要注意下列項 目:

- 編輯 debian/control 來分解套件 version (http://www.debian.org/doc/ debian-policy/ch-controlfields#s-f-Version).
- 檢查建立套件的 scripts 並使用 CFLAGS=-g -Wall 來編譯軟體。
- 設定 DEB\_BUILD\_OPTIONS=nostrip, noopt 環境變數來建立 Debian 套件。

請參閱 Policy 10.1 (http://www.debian.org/doc/debian-policy/ch-files#s10.1) 了解更多資訊。

#### 檢查函式庫相依關係

使用 1dd 可查看程式相依的函式庫:

```
$ ldd /bin/ls
    librt.so.1 => /lib/librt.so.1 (0x4001e000)
    libc.so.6 => /lib/libc.so.6 (0x40030000)
    libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
    /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

所以在 chroot 的環境下要使用 1s,上列的函式庫在你的 chroot 的環境中都必需可用才行。

下面的命令也很有用:

- strace: 追蹤系統呼叫和訊息 (signal)
- ltrace: 追蹤函式庫呼叫

#### 用記憶體漏失偵測工具來進行除錯

Debian 中有幾個記憶體漏失偵測工具。

- njamd
- valgrind
- dmalloc
- electric-fence
- memprof
- memwatch(沒有包成套件,可在 memwatch(http://directory.fsf.org/devel/ debug/memwatch.html)取得。)
- mpatrol
- leaktracer
- libgc6
- Insure++ from Parasoft (http://www.parasoft.com)。(非自由軟體,商業用途要 付費)

也可參閱 Debugging Tools for Dynamic Storage Allocation and Memory Management (http: //www.cs.colorado.edu/homes/zorn/public\_html/MallocDebug.html)。

# 13.7.3 Flex – 更好的 Lex

flex 是一個快速的字詞分析產生器。

flex 的參考資料:

- info flex (tutorial)
- flex(1)

需要提供你自己的 main() 和 yywrap() ,或者你的 program.1 在不用 library 時應該看起 來看像這樣 (yywrap是一個巨集; % option main 隱含地打開了 % option noyywrap):

```
%option main
%%
.|\n ECHO;
%%
```

另外,還可以在 cc 命令列末尾加上 -1f1 連接器的選項來進行編譯 (就像 AT&T-Lex 使用 -11 一樣),此時就不需要 % option 了。

## 13.7.4 Bison - 更好的 Yacc

Debian 中有幾個套件提供了與 Yacc 兼容的 LALR 文法分析生成器:

- bison: GNU LALR 文法分析產生器
- byacc: The Berkeley LALR 文法分析產生器
- btyacc: Backtracking parser generator ,以 byacc 爲基礎

bison 的参考資料:

- info bison (tutorial)
- bison(1)

需要提供自己的 main() 和 yyerror() 。 main() 呼叫 yyparse() , 而 yyparse() 呼叫 yylex() , 通常由 FleX 建立。

<u> ୧</u>୧

00 00

#### 13.7.5 Autoconf

autoconf 一個 shell script 生成工具,由它生成的 script 能自動設定軟體的源碼套件,以適用 於各種使用完整 GNU build 系統的 UNIX-like 系統。

autoconf 會產生設定 script 的 configure。 configure 使用 Makefile.in 模版來自動 建立適合的 Makefile。

#### 编譯和安裝程式

Debian 不會改動 /usr/local/下的文件 (參閱 '多樣性支援' 19)。所以如果是從源碼來編譯程式,並將其安裝到 /usr/local/下,就不會影響到 Debian。

\$ cd src
\$ ./configure --prefix=/usr/local
\$ make
\$ make install # 這會把檔案放到系統中

#### 反安裝程式

如果你仍保有源碼,而且是用 autoconf/automake,你也記得是如何進行設定的話:

\$ ./configure all-of-the-options-you-gave-it

# make uninstall

另一種方法是,如果可以確定安裝過程將檔案都放在 /usr/local/, 並且該目錄下沒有什麼 別的重要文件的話,可用下面的命令將其全部刪除:

# find /usr/local -type f -print0 | xargs -0 rm -f

如果不能確定文件安裝到什麼位置,最好使用 checkinstall,可提供明確用來反安裝的路徑。

# 13.8 Web

基本的交談性動態網頁可以利用下列方式來實現:

- 查詢(query)使用 HTML 格式呈現在使用者的瀏覽器上。
- 填寫和點選網頁上的項目會送出編碼後的參數到網址。1從瀏覽器到網頁伺服器的過程,
   例如:
  - http://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3
  - http://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3
  - http://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3
- 網頁伺服器上的 CGI 程式 (任何形式的檔案 program.\*) 會收到解碼後的參數 "VAR1=VAL1 VAR2=VAL2 VAR3=VAL3"來當成環境變數 "QUERY\_STRING" 並執行。
- CGI 程式的 stdout 將會送給使用者的瀏覽器以呈現出交談性的動態網頁。

爲了安全性的考量,最好能避免手動分析這些 CGI 參數。有一些 Perl(參閱'Perl' 209)和 Python(參閱'Python' 210)能達成這些功能。PHP (http://www.php.net/) 已經內建這些 函式。如果需要儲存 client 端的資料,請使用 cookies。處理 client 端的資料,通常是使用 javascript。

更多的資訊請閱讀 The Common Gateway Interface (http://hoohoo.ncsa.uiuc. edu/cgi/), The Apache Software Foundation (http://www.apache.org/), 和 JavaScript (http://www.mozilla.org/js/)。

在瀏覽器上利用已編碼的網址 http://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutorial 在 Google 上查詢 "CGI tutorial" 來查閱 CGI script 對於 Google 伺服器的處理是不錯的辦法 喔。

# 13.9 文件準備

#### 13.9.1 roff 排版

傳統上, roff 是主要的 Unix 文字處理系統。

參閱 roff(7) , groff(7) , groff(1) , grotty(1) , troff(1) , groff\_mdoc(7) , groff\_man(7) , groff\_ms(7) , groff\_me(7) , groff\_mm(7) 和 info groff 。

在 -me 巨集中有份不錯的教程。如果你是用 groff (1.18 以上版本),找到 /usr/share/doc /groff/meintro.me.gz 並做下面的動作:

\$ zcat /usr/share/doc/groff/meintro.me.gz | \
 groff -Tascii -me - | less -R

下面的作法將產生一份完整的純文字檔案:

\$ zcat /usr/share/doc/groff/meintro.me.gz | \
 GROFF\_NO\_SGR=1 groff -Tascii -me - | col -b -x > meintro.txt

<sup>&</sup>lt;sup>1</sup>這裡的 %nn 是將字元編碼成 16進位。 nn.

如果要列印的話,就用 PostScript 輸出。

\$ groff -Tps meintro.txt | lpr \$ groff -Tps meintro.txt | mpage -2 | lpr

#### 13.9.2 SGML

準備工作:

# apt-get install debiandoc-sgml debiandoc-sgml-doc

debiandoc-sgml 的參考資料:

- /usr/share/doc/debiandoc-sgml-doc
- debiandoc-sgml(1)
- DocBook: The Definitive Guide (/usr/share/doc/docbook-defguide/html/ docbook.html), by Walsh and Muellner, (O'Reilly) (package docbook-defguide)

SGML 能夠管理一份文件的多種格式。 Debiandoc 是個較簡單的 SGML 系統,本文件就是用 它來編寫。只需將原本的文字檔對下列字元進行少許轉換:

- "<" -> &lt;
- ">" -> >
- ""-> (不可斷行的空白)
- "&" -> & amp;
- "%" -> %
- "(c)" -> & copy;
- "-"-> –
- "---" -> —

要把一段文字變成不輸出的註解的話,請輸入:

<!-- State issue here  $\dots$  -->

要把一段文字變成可選擇是否輸出的註解的話,請輸入:

<![ %FIXME; [ State issue here ... ]]>

在 SGML 中,只有首次定義 (first definition)是有效的。例如:

```
<!entity % qref "INCLUDE">
<![ %qref; [ <!entity param "Data 1"> ]]>
<!entity param "Data 2">
&param;
```

最後結果是 "Data 1"。如果第一行是用 "IGNORE" 而不是 "INCLUDE",則結果是 "Data 2" (第二行是條件判斷式)。而且,重覆出現的文字可提前在文件中定義。

```
<!entity whoisthis "my">
Hello &whoisthis; friend.
This is &whoisthis; book.
```

結果如下:

Hello my friend. This is my book.

可 參 閱 examples (http://www.debian.org/doc/manuals/debian-reference/ examples/) 目錄中簡短的 SGML 範例文件 sample.sgml。

當 SGML 文件不斷增大後,有時以 Tex 做為後端文字處理會出現問題,請參閱 'TeX/LaTeX' 219。

#### 13.9.3 TeX/LaTeX

事前準備:

```
# tasksel # 請選擇 Miscellaneous --> TeX/LaTeX environment
```

LaTeX 的参考手册:

- The teTeX HOWTO: The Linux-teTeX Local Guide (http://www.tldp.org/HOWTO/ TeTeX-HOWTO.html)
- tex(1)
- latex(1)
- *The TeXbook*, by Donald E. Knuth, (Addison-Wesley)<sup>2</sup>
- LaTeX A Document Preparation System, by Leslie Lamport, (Addison-Wesley)
- *The LaTeX Companion*, by Goossens, Mittelbach, Samarin, (Addison-Wesley)

TeX 是一個強大的寫作環境。許多 SGML 處理程式使用它來當作後端文字處理器。1yx、1yx-xforms或1yx-qt內附的 Lyx 以及 GNU TeXmacs 內附的texmacs提供了相當不錯的 WYSIWYG 編輯環境。使用者大多採用 Emacs 和 Vim 當作撰寫 LaTeX 原始碼的編 輯器。

下列有許多線上參考資料:

 teTeX - A Documentation Guide (/usr/share/doc/texmf/newhelpindex.html) (tetex-doc package)

<sup>&</sup>lt;sup>2</sup>可以到 ftp://ftp.dante.de/pub/tex/systems/knuth/tex/texbook.tex 取得書中的 Tex 原始碼。ftp://ftp.dante.de/pub/tex/systems/knuth/lib/manmac.tex内建大部分所必須的巨集。加入\input manmac \proofmodefalse來註解掉第七到第十的內容,就可以使用tex來編譯該文件了。 建議買下此書(以及 Donald E. Knuth 其他的書籍)來替代線上的版本,即使是 Tex 原碼對於 TeX 輸入是很棒的範例。

- A Quick Introduction to LaTeX (http://www.msu.edu/user/pfaffben/ writings/)
- A Simple Guide to Latex/Lyx (http://www.stat.rice.edu/~helpdesk/howto/ lyxguide.html)
- Word Processing Using LaTeX (http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex\_basic/latex\_basic.html)
- Local User Guide to teTeX/LaTeX (http://supportweb.cs.bham.ac.uk/ documentation/LaTeX/lguide/local-guide/local-guide.html)

當文件越來越大時,編譯 TeX 有時候會出錯。使用者必須編輯/etc/texmf/texmf.cnf (或 者適當地修改 /etc/texmf/texmf.d/95NonPath並執行update-texmf)加大 pool size 來 修正該問題。

#### 13.9.4 文學編程

和在編程時寫文件不同的,文學編程者是在寫文件時編程。這個方法確保程式有個良好的文件。

要對文學編程有更多了解,請參閱 Literate Programming (http://www.literateprogramming.com/)。

#### Noweb

準備工作:

# apt-get install nowebm

Noweb 的參考資料:

- Noweb A Simple, Extensible Tool for Literate Programming (http://www.eecs. harvard.edu/~nr/noweb/)
- noweb(1)

這是個 WEB-like 的文學編程工具,可更容易地提供擴充性和與語言無關性,<sup>3</sup> When noweb is invoked, it writes the program source code to the output files mentioned in the noweb file, and it writes a TeX file for typeset documentation.

Debian 中的 if updown 套件是個好例子。

- \$ apt-get source ifupdown
- \$ cd ifupdown\*
- \$ make ifupdown.pdf ifupdown.ps

<sup>&</sup>lt;sup>3</sup>This WEB has **nothing** to do with World Wide Web。WEB (用 PASCAL) 和 CWEB (用 C/C++) 是傳統的文學 編程工具。

#### Doxygen

準備工作:

# apt-get install doxygen doxygen-doc doxygen-gui

Doxygen 的參考資料 (也是用 doxygen 建立的!):

- Homepage (http://www.doxygen.org/)
- /usr/share/doc/doxygen-doc/html/index.html

它可以由 C++、 C、 Java、 IDL 和一些 PHP 和 C# 程式產生 HTML、 RTF、 Unix manual pages、 PostScript 和 PDF (使用 LaTeX) 文件。Doxygen 相容於 JavaDoc (1.1)、 Qt-Doc、 KDOC 並被特別設計爲用在使用到 Troll Tech's Qt (http://www.trolltech.no/qt/) toolkit 的專案中。它可以建立 include 相依圖, 合作圖表, 和圖型化的類別繼承圖, 即使在沒有被文件化的程式。它的輸出是類似 Qt 的文件。

# 13.10 包裝套件

準備工作:

包裝套件的參考資料:

- 'Debian 套件管理系統' 10 (basics)
- Debian New Maintainers' Guide (tutorial)
- dh-make(1)
- Debian Developer's Reference (best practice)
- Debian Policy Manual (authoritative)
- Packaging Manual (Potato)

# 13.10.1 包装一個單一二進位套件

Joey Hess 的單一二進位套件快速粗糙的打包法:

```
# mkdir -p mypkg/usr/bin mypkg/DEBIAN
# cp binary mypkg/usr/bin
# cat > mypkg/DEBIAN/control
Package: mypackage
Version: 1
Architecture: i386
```

Maintainer: Joey Hess <joeyh@debian.org>
Description: my little package
Don't expect much.
^D
# dpkg-deb -b mypkg

## 13.10.2 使用工具來進行包裝工作

使用 dh-make 套件中的 dh\_make 來建立一個基本的套件。接著按照 dh-make (1) 中描述的 方法進行。會用到 debian/rules 中的 debhelper 。

一個較舊的方法是使用 debmake 套件中的 deb-make 。不需要 debhelper script,只需要 shell 就行了。但請不要再使用這樣的方法了。

有關多重源碼套件 (multiple-source) 的例子,參閱"mc" (dpkg-source -x mc\_4.5.54.dsc),其中用到 Adam Heath(<doogie@debian.org>)的"sys-build.mk" 以及"glibc" (dpkg-source -x glibc\_2.2.4-1.dsc),它使用到已故的 Joel Klecker(<espy@debian.org>)所寫的另一個系統。

# **Chapter 14**

# GnuPG

參考資料:

- gpg(1).
- /usr/share/doc/gnupg/README.gz
- /usr/share/doc/gnupg-doc/GNU\_Privacy\_Handbook/中的 GNU privacy handbook (安裝 gnupg-doc 套件)

# 14.1 安裝 GnuPG

#	gpggen-key	#	產生新的 key
#	gpggen-revoke <i>my_user_ID</i>	#	產生 my_user_ID 的廢止用 key
#	<pre>host -l pgp.net   grep www less</pre>	#	指出 pgp keyservers

加入一個預設的穩定 keyserver 到\$HOME/.gnupg/gpg.conf檔案内:

keyserver hkp://subkeys.pgp.net

必須注意不能建立 2 個以上的 sub-keys,如果你這樣做,pgp.net 上的 keyservers 會廢除 (corrupt) 你的 key。使用新版的 gnupg (>1.2.1-2) 來處理這些廢除的 subkeys。參閱 http: //fortytwo.ch/gpg/subkeys。

# 14.2 使用 GnuPG

檔案處理:

```
$ gpg [options] command [args]
$ gpg {--armor|-a} {--sign|-s} file # 將 file 簽 名 , 存 至 文 字
檔 file.asc
```

# 14.3 管理 GnuPG

Key 管理:

```
$ gpg --edit-key user_ID
$ gpg -o file --exports
                                         # "help",互動式的説明
                                        # 將所有的 key 匯出至 file
                                         # 匯入 file 中所有的 key
   $ gpg --imports file
  $ gpg --send-keys user_ID
                                          # 將 user ID 的 key 傳
                                                                     送
到 keyserver
  $ gpg --recv-keys user_ID
                                          # 從 keyserver 接
收 user_ID 的 key
  $ gpg --list-keys user_ID
$ gpg --list-sigs user_ID
                                         # 列出 user_ID 的 key
                                         # 列出 user_ID 的 sig.
  $ gpg --check-sigs user_ID # 檢查 user_ID 的 sig.
$ gpg --fingerprint user_ID # 檢查 user_ID 的 fingerprint
   $ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
    | awk '{print $2}' | sort -u | xargs gpg --recv-keys # 取得未知
的 keys
    # 將所有未知的 sigs 更新 key 。
                                           # 更新本地端的 keyrin。
   $ qpq --refresh-keys
```

Trust code:

No ownertrust assigned / not yet calculated.
 e 計算信任度失敗。
 q 計算所需的資訊不足。
 n 永不信任這把 key。
 m Marginally trusted.
 f 完全信任 (Fully trusted)。
 u 極度信任 (Ultimately trusted)。

下面的命令會將我的 key "A8061F32" 上傳到熱門的 key servers hkp://subkeys.pgp.net:

\$ gpg --keyserver hkp://subkeys.pgp.net --send-keys A8061F32

# 14.4 在應用程式中使用 GnuPG

# 14.4.1 在 Mutt 中使用 GnuPG

在~/.muttrc 中加入下列的内容,则如果你在 index 選單中輸入 'S' 的話,將會自動啓動一個低速的 GnuPG。

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

# 14.4.2 在 Vim 中使用 GnuPG

```
將 examples subdirectory (http://www.debian.org/doc/manuals/
debian-reference/examples/) 中的 _vimrc 文件的内容加到 ~/.vimrc 就可以自
然的執行 GnuPG。
```

# Chapter 15

# Debian 的技術支援

下列資源提供了與 Debian 相關的幫助、建議和支援。在郵件列表裡大呼救命之前,務必先好好使用這些資源自助。:)

注意,系統中就有大量的文件,可使用 WWW 瀏覽器來讀取,可透過 dwww 或 dhelp 指令在相關的套件中找到它們。

# 15.1 參考資料

下列資源通常適用於 Debian 和 Linux。如果它們的內容彼此間出現衝突,應該相信首要的 (primary) 來源而不是像本文件之類的次要 (secondary) 來源。

- 安裝手冊 (Installation Manual) (首要的)
  - 在安裝和升級前閱讀。
  - 網站: http://www.debian.org/releases/stable/installmanual
  - 網站: http://www.debian.org/releases/testing/installmanual (正 在進行後,有時會不存在)
  - 套件: Not available in install-doc: Bug#155374
  - 檔案: DebianCDunder/doc/
- 發佈注意事項 (Release Notes) (首要的)
  - 就算你很有經驗,在安裝和升級前也一定要讀。
  - 網站: http://www.debian.org/releases/stable/releasenotes
  - 網站: http://www.debian.org/releases/testing/releasenotes (正在 進行後,有時會不存在)
  - 套件: Not available in install-doc: Bug#155374
  - 檔案: DebianCDunder/doc/
- FAQ (次要的)
  - 常見問題
  - 網站: http://www.debian.org/doc/manuals/debian-faq/

- 套件: doc-debian
- 檔案: /usr/share/doc/debian/FAQ/index.html
- Debian 參考手冊 (Debian Reference) (次要的)
  - 最全面性的使用手册
  - 網站: http://www.debian.org/doc/manuals/debian-reference/
  - 套件: debian-reference-zh-tw
  - 檔案: /usr/share/doc/Debian/reference/
- APT HOWTO (次要的)

\_ 網

- Debian 套件管理系統的詳細使用指南。(woody)
- 網站: http://www.debian.org/doc/manuals/apt-howto/
- 套件: apt-howto
- 檔案: /usr/share/doc/Debian/apt-howto/
- Debian 安全手冊 (Securing Debian Manual) (次要的)

:

- 有關如何強化預設安裝的 Debian 系統安全的詳細使用指南。(woody)
  - http://www.debian.org/doc/manuals/
- securing-debian-howto/

站

- 套件: harden-doc
- 檔案: /usr/share/doc/harden-doc/html/securing-debian-howto/
- dselect 初學者文件 (dselect Documentation for Beginners) (次要的)
  - dselect 的初學教程。
  - 網 站 : http://www.debian.org/releases/woody/i386/ dselect-beginner
  - 套件: Not available in install-doc: Bug#155374
  - 檔案: DebianCDunder/doc/
- Debian Policy Manual (首要的)
  - Debian 的基本技術架構。
  - 網站: http://www.debian.org/doc/debian-policy/
  - 套件: debian-policy
  - 檔案: /usr/share/doc/debian-policy/
- Debian 開發人員參考手冊 (Debian Developer's Reference) (首要的)
  - 開發人員需要了解的基礎知識。
  - 我們之中的一部分人也需要讀一讀。
  - 網站: http://www.debian.org/doc/manuals/developers-reference/
  - 套件: developers-reference
  - 檔案: /usr/share/doc/developers-reference/
- Debian 新維護者指南 (Debian New Maintainers' Guide) (首要的)
  - 開發人員的實用指南。
  - 我們中的一部分需要閱讀其中的包裝套件教程。
  - 網站: http://www.debian.org/doc/manuals/maint-guide/
  - 套件: maint-guide

- 檔案: /usr/share/doc/maint-guide/
- Packaging Manual (Potato)
   Potato 中的 packaging-manual 套件。(已移到 Developer's Reference 的附錄中。)
- Unix-style manual pages (首要的)
  - dlocate -man package-name (list available)
  - man section command-name
- GNU-style info pages (首要的)
  - info (access top level)
  - info command-name
- Package specific documents (首要的)
   在 /usr/share/doc/package-name 可以找得到
- LDP: Linux Documentation Project (次要的)
  - 一般通用的 Linux HOWTOs 和 mini-HOWTOs 。
  - 網站: http://www.tldp.org/
  - 套件: doc-linux-text 和 doc-linux-html
  - 檔案: /usr/share/doc/HOWTO/
- Linux Gazette (次要的) 每月發行
  - The Linux Gazette
  - 網站:http://www.linuxgazette.com/
  - 套件: lg-all 或 lg-latest-two
  - 檔案: /usr/share/doc/lg/
- DDP: Debian Documentation Project (次要的)
  - Debian-specific manuals
  - 網站: http://www.debian.org/doc/
- Debian Developers' Corner (次要的)
  - Debian 開發者的重要資訊。
  - 一般使用者可以用來增廣見識。
  - 網站: http://www.debian.org/devel/
- 源碼 (絕對是首要的)
  - 沒人會表示反對吧:-)
  - 按照'源碼' 10 的方法下載源碼。
- Internet Assigned Numbers Authority (主要的)
  - 網站:http://www.iana.org/
  - 套件: doc-iana
  - 檔案: /usr/share/doc/doc-iana/
- Internet requests for comments (IETF 標準) (主要的)
  - 網站: http://www.ietf.org/rfc.html

- 套件: doc-rfc
- 檔案: /usr/share/doc/RFC/

下列是 Unix 通用的參考資源。注意各種 Unix 系統之間存在著稍許不同。設備名稱和初始化方式需要格外注意。

- The UNIX Programming Environment
  - 閱讀本書以了解 UNIX 如何運行。
  - 作者: B. W. Kernighan 和 R. Pike
  - 由 Princeton Hall Software Series 發行
- The C Programming Language (第二版)
  - 閱讀本書來學習 ANSI C。
  - 作者: B. W. Kernighan 和 D. M. Ritchie
  - 由 Princeton Hall Software Series 發行
- UNIX Power Tools
  - 閱讀本書來學習 Unix 使用技巧。
  - 作者: Jerry Peek, Tim O'Reilly 和 Mike Loukides
  - 由 O'Reilly and Associates 發行
- Essential System Administration (第二版)
  - 閱讀本書來學習如何管理各種風格的 Unix 系統。
  - 作者: Aeleen Frisch
  - 由 O'Reilly and Associates 發行
- Linux: Rute User's Tutorial and Exposition
  - 管理 GNU/Linux 系統的精裝版線上書籍。
  - 作者: Paul Sheer
  - 由 Prentice Hall 發行
  - 套件: rutebook (從 non-free)
  - 檔案: /usr/share/doc/rutebook/
- Bell Labs: Computing Sciences Research
  - 有關 Unix 歷史的豐富文獻
  - 主要: http://cm.bell-labs.com/cm/cs/
  - 技術報告精選: http://cm.bell-labs.com/cm/cs/cstr.html
  - 一些論文: http://cm.bell-labs.com/cm/cs/papers.html
- 線上 Linux 通用技術支援資料
  - Debian Planet (http://www.debianplanet.org/)
  - debianHELP(http://www.debianhelp.org/)
  - Linux.com (http://linux.com/)
  - The Linux Home Page at Linux Online (http://www.linux.org/)
  - Red Hat (commercial Linux vender) (http://www.redhat.com/) (RPM, Sys-V init)

- SuSE, Inc. (commercial Linux vender) (http://www.suse.de/) (RPM, Sys-V init)
- Slackware (http://www.slackware.com/) (TGZ, BSD-style init)
- 線上一般的 Unix 指南及資源
  - The UNIX System by The Open Group (http://www.unix.org/)
  - A UNIX Introductory Course from Ohio State University (http://www-wks.acs. ohio-state.edu/unix\_course/unix.html)
  - UNIXhelp from The University of Edinburgh (http://unixhelp.ed.ac.uk/)
  - Unix / Programming Information (http://arioch.unomaha.edu/~jclark/ #info)
  - comp.unix.questions FAQ (http://www.faqs.org/faqs/unix-faq/faq/)
  - comp.unix.user-friendly FAQ (http://www.camelcity.com/~noel/usenet/ cuuf-FAQ.htm)
  - FreeBSD Documentation (http://www.freebsd.org/docs.html)
  - The FreeBSD Handbook (http://www.freebsd.org/doc/en\_US. IS08859-1/books/handbook/index.html)
  - UNIX GUIDE (http://ieee.uow.edu.au/documents/)
  - The Unix Heritage Society (http://www.tuhs.org/)
- 自由軟體專案首頁
  - GNU Project (http://www.gnu.org/)
  - The Linux Documentation Project (http://www.tldp.org/)
  - The Linux Kernel Archives (http://www.linux.org/)
  - The XFree86 Project, Inc (http://www.xfree86.org/)
  - GNOME (http://www.gnome.org/)
  - K Desktop Environment (http://www.kde.org/)
  - GNU software at Red Hat (http://sources.redhat.com/)
  - Mozilla (http://www.mozilla.org)
  - FreeBSD (http://www.freebsd.org/)
  - OpenBSD (http://www.openbsd.org/)
  - NetBSD (http://www.netbsd.org/)

# 15.2 查詞意

Debian 中使用了大量術語和縮寫,下面的命令將會回答你大多數的疑問(需安裝dict和相關套件):

\$ dict put-a-weird-word-here

# 15.3 查找流行的 Debian 套件

Debian 中有許多套件,有時很難決定該裝哪一個。參閱 Debian Popularity Contest Results (http://popcon.debian.org/) 可了解別人都在用什麼軟件。亦可安裝 popularity-contest 套件參加投票。

# 15.4 Debian 錯誤追蹤系統

Debian distributoin 有一個 bug tracking system (BTS) (http://bugs.debian.org/),它將 來自使用者和開發人員的錯誤報告的詳細內容進行歸檔,每個錯誤都有一個編號,錯誤報告將 一直存在於資料庫中,直到獲得已更正的標記。

在發送錯誤報告之前,請先檢查一下別人是否已提出了相同的錯誤報告。 World Wide Web (http://bugs.debian.org/)或其它地方 elsewhere (http://www.debian.org/Bugs/Access) 均可找到目前未更正錯誤列表。參閱 '檢查 Debian 中的 bugs 並尋求幫助' 76。

許多嚴重錯誤的報告標記爲 FTBFS。意思是 "Fails To Build From Source"。

http://www.debian.org/Bugs/Reporting 中描述了錯误報告方法。

# 15.5 郵件列表

閱讀最新的"debian-devel-announce"(英文,唯讀且低流量)與 Debian 保持同步。

Debian 使用者最感興趣的郵件列表是 "debian-user" (英文,開放且高流量) 和其他的 "debian-user-language" 郵件列表 (給其他語言使用者)。

想了解這些郵件列表的詳細資訊和如何訂閱,請參閱 http://lists.debian.org/。發言前請先搜尋答案,並注意遵守郵件列表相關禮儀和規則。

如果你不希望在回覆至郵件列表時收到複本的話,你應該用 Mail-Followup-To:的標頭, 很有效的。這是郵件列表中非官方的習慣,在 http://cr.yp.to/proto/replyto.html 有些說明。

# 15.6 Internet Relay Chat (IRC)

IRC (Internet Relay Chat) 可讓你與世界各地的人進行即時地聊天。在 freenode (http://www.freenode.info/) 上可找到與 Debian 相關的 IRC 頻道。要連上頻道,你需要一個 IRC 客户端。一些流行的客户端有 XChat、BitchX、ircII、irssi、epic4 和 KSirc,每一個都可以在 Debian 套件中找到。安裝好客户端後,你需要連接到伺服器。對大多數的客户端,可輸入:

/server irc.debian.org

當連接上伺服器後,可輸入下面的指令加入 #debian 頻道:

/join #debian

要離開 #debian 頻道的話,就輸入

/part #debian

要離開 irc 的軟體,就輸入

/quit

要送個私人的訊息"Hello Mr. Foo"給foo,請輸入

/msg foo Hello Mr. Foo

要注意的是,你所打的任何東西只要不是以 / 開頭的話,就會被當做是訊息而送至頻道上。 注意:像 XChat 這類的客户端在加入伺服器/頻道的圖形使用者介面會有不同。

# 15.7 搜尋引擎

有許多搜尋引擎提供有關 Debian 的文件:

- Debian WWW search site (http://search.debian.org/).
- Google (http://www.google.com/): 將 "site:debian.org" 加入搜尋字詞中。
- Google Groups (http://groups.google.com/): 新聞群組搜尋引擎。將 "group:linux.debian.\*"加入搜尋字詞中。
- AltaVista (http://www.altavista.com/)

例如,搜尋字串 "cgi-perl" 會得到更多關於這個套件的詳細說明,而不只是在 control 檔中的 簡要說明。請參考 '檢查 Debian 中的 bugs 並尋求幫助' 76 獲得相關建議。

## 15.8 網站

下面是我收集的一些特定主是討論的零散 URL。

- IBM developerWorks: Linux (http://www.ibm.com/developerworks/linux/)
- Adrian Bunk's latest packages (back port to stable) (http://www.fs.tum.de/ ~bunk/)
- Linux on Laptops (http://www.linux-laptop.net/)
- Xterm FAQ (http://dickey.his.com/xterm/xterm.faq.html)
- EXT3 File System mini-HOWTO (http://www.zip.com.au/~akpm/linux/ext3/ ext3-usage.html)

- Large File Support in Linux (http://www.suse.de/~aj/linux\_lfs.html)
- Window Managers for X (http://www.xwinman.org)
- Linux USB Project (http://www.linux-usb.org/)
- SuSE pages for CJK (http://www.suse.de/~mfabian/suse-cjk/suse-cjk. html)
- LNX-BBC (Business-card-sized boot CD project) (http://www.lnx-bbc.org/)
- Linux info by Karsten Self (partitioning, backup, browsers...) (http://kmself.home. netcom.com/Linux/index.html)
- Backup info HOWTO by Alvin Oga (http://www.Linux-Backup.net/)
- Security info HOWTO by Alvin Oga (http://www.Linux-Sec.net/)
- Various UNOFFICIAL sources for APT (http://www.apt-get.org/)
- Laptop Ethernet Configuration (http://www.orthogony.com/gjw/lap/ lap-ether-intro.html)
# Appendix A

## 附錄

## A.1 作者

Debian 參考手冊 是由 青木 修 (Osamu Aoki) < osamu \#at \#debian.org> 所發起,剛開始 時是做爲個人的安裝備忘錄,而後叫做 "Quick Reference ..."。許多內容來自於 "debian-user" 郵件列表的彙整庫。也參考了 "Debian Installation Manual" 和 "Debian Release Notes"。

而後由 Debian Documentation Project (http://www.debian.org/doc/ddp) (DDP) 的積極 參與者、也是目前"The Debian FAQ"的主要維護者 Josip Rodin 的建議,這份文件更名為 "Debian 參考手冊",並和"The Debian FAQ"中的幾個類似參考的章節合併。而"Debian 快 速參考手冊"則是以節錄的方式形成。

本文件是由下列 QREF 團隊成員所編輯,翻譯和擴充:

- 原本 "Quick Reference..." 的英文原版
  - 青木 修 (Osamu Aoki) < osamu \ #at \ #debian.org> (所有内容的領導者)
- 英文版校對和後續的貢獻
  - Esko Araj 酺vi <edu\#at\#iki.fi> (etch的更新)
  - Thomas Hood < jdthood \#at \#yahoo.co.uk> (網路相關的內容)
  - Brian Nelson <nelson\#at\#bignachos.com> (特別是X相關的内容)
  - David Sewell <dsewell \#at \#virginia.edu>(已退休)
  - Jan Michael C Alonzo < jmalonzo \#at \#spaceants.net>
  - Daniel Webb <webb\#at\#robust.colorado.edu>
  - 所有翻譯人員的回饋
- 法文版翻譯
  - Guillaume Erbs <gerbs\#at\#free.fr>(法文版領導者)
  - Renald Casagraude <rcasagraude \#at \#interfaces.fr>
  - Jean-Pierre Delange <adeimantos\#at\#free.fr>
  - Daniel Desages <daniel\#at\#desages.com>
- 義大利文版翻譯
  - Davide Di Lazzaro <mc0315\#at\#mclink.it> (義大利文版領導者)
- 葡萄牙文版翻譯
  - Paulo Rogerio Ormenese <pormenese \#at \#uol.com.br>(葡萄文版領導者)
  - Andre Luis Lopes <andrelop\#at\#ig.com.br>

- Marcio Roberto Teixeira <marciotex\#at\#pop.com.br>
- Rildo Taveira de Oliveira <to\_rei\#at\#yahoo.com>
- Raphael Bittencourt Simoes Costa <raphael-bsc\#at\#bol.com.br>
- Gustavo Noronha Silva <kov\#at\#debian.org> (coordinator)
- 西班牙文版翻譯
  - Walter Echarri <wecharri \#at \#infovia.com.ar> (西班牙文版領導者)
  - Jose Carreiro <ffx\#at\#urbanet.ch>
- 德文版翻譯
  - Jens Seidel <tux-master\#at\#web.de> (德文版領導者)
  - Willi Dyck <wdyck\#at\#gmx.net>
  - Stefan Schroeder <stefan\#at\#fkp.uni-hannover.de>
  - Agon S. Buchholz <asb\#at\#kefk.net>
- 波蘭文版翻譯 下列是 PDDP (http://debian.linux.org.pl) 的成員:
  - Marcin Andruszkiewicz
  - Mariusz Centka <mariusz.centka\#at\#debian.linux.org.pl>
  - Bartosz Feński < fenio \#at \#debian.linux.org.pl>(波蘭文版領導者)
  - Radosław Grzanka <radekg\#at\#debian.linux.org.pl>
  - Bartosz 'Xebord' Janowski
  - Jacek Lachowicz
  - Rafał Michaluk
  - Leonard Milcin, Jr.
  - Tomasz Z. Napierała <zen\#at\#debian.linux.org.pl>
  - Oskar Ostafin <cx\#at\#debian.linux.org.pl>
  - Tomasz Piękoś
  - Jacek Politowski
  - Mateusz Prichacz <mateusz \#at \#debian.linux.org.pl>
  - Marcin Rogowski
  - Paweł Różański
  - Mariusz Strzelecki
  - Krzysztof Ścierski
  - Przemysław Adam Śmiejek <tristan\#at\#debian.linux.org.pl>
  - Krzysztof Szynter
  - Mateusz Tryka <uszek\#at\#debian.linux.org.pl>
  - Cezary Uchto
  - Krzysztof Witkowski <tjup\#at\#debian.linux.org.pl>
  - Bartosz Zapałowski <zapal\#at\#debian.linux.org.pl>
- 簡體中文翻譯
  - Hao "Lyoo" LIU<iamlyoo\#at\#163.net>
  - Ming Hua <minghua\#at\#rice.edu>
  - Xiao Sheng Wen <atzlinux\#at\#163.com> (leader: zh-cn)
  - Haifeng Chen <optical.dlz\#at\#gmail.com>
  - Xie Yanbo <xieyanbo\#at\#gmail.com>
  - easthero <easthero \#at \#gmail.com>
- 繁體中文翻譯
  - 葉信佑 (asho) <asho\#at\#debian.org.tw> (繁體中文版領導者)
  - 唐偉清 (wctang) <wctang\#at\#csie.nctu.edu.tw>

- 日文翻譯
  - Shinichi Tsunoda <tsuno\#at\#ngy.1st.ne.jp>(日文版領導者)
  - Osamu Aoki <osamu\#at\#debian.org>
- 芬蘭翻譯
  - Esko Araj酺vi <edu\#at\#iki.fi>(芬蘭版領導者)

QREF 是原本文件標題 "Quick Reference..." 的縮寫,也是在 qref.sourceforge.net 上的專案名稱。

在編寫本文件時,很多在 Debian 系統中的 manual 頁面和 info 頁面被當做是主要的參考 資料。 To the extent Osamu Aoki considered within the fair quotation stature, many parts of them, especially command definitions, were used as phrase pieces after careful editorial efforts to fit them into the style and the objective of this document.

在 'Debian 基礎' 5 中大多數的內容來自於 "The Debian FAQ" (2002 年 3 月):

- 5. The Debian FTP archives: ftparchives.sgml (整個章節)
- 6. Basics of the Debian Package Management System: pkg\_basics.sgml (整個章節)
- 7. The Debian Package Management Tools: pkgtools.sgml (整個章節)
- 8. Keeping Your Debian System Up To Date: uptodate.sgml (整個章節)
- 9. Debian and the kernel: kernel.sgml (整個章節)
- 10. Customizing your installation of Debian GNU/Linux: customizing.sgml(部分章 節)

這些 "The Debian FAQ" 的章節為了反應在 Debian 系統的變化,是經過了較大的重組後才加入本文件的。現在這兩份文件的內容都是同時被更新的。

最初 "Debian FAQ" 是由 J. H. M. Dassen (Ray) 和 Chuck Stickelman 建立和維護。重新編寫過的 "Debian FAQ" 作者是 Susan G. Kleinmann 和 Sven Rudolph 。在他們之後, "The Debian FAQ" 是由 Santiago Vila 來維護。目前的維護者是 Josip Rodin 。

"The Debian FAQ" 部分的資訊是來自於:

- The Debian-1.1 release announcement ,由 Bruce Perens (http://www.perens.com/) 所著。
- The Linux FAQ , 由 Ian Jackson (http://www.chiark.greenend.org.uk/ ~ijackson/)所著。
- Debian Mailing List Archives (http://lists.debian.org/) ,
- the dpkg programmers' manual 和 the Debian Policy manual (參閱 '參考資料' 227)
- 眾多開發人員、自願者、beta版測試人員,和
- 其作者的片斷記憶。:-)

在"Tutorial" 這一節中有部分內容是來自於:

- "Debian Tutorial" by Havoc Pennington, Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer, and Ivan E. Moore II. (這份文件是承接自"Linux User's Guide" by Larry Greenfield 。)
- "Debian GNU/Linux: Guide to Installation and Usage" by John Goerzen and Ossama Othman.

作者在此感謝所有曾經幫助完成本文件的人們。

#### A.2 保證

因爲我不是專家,我也不敢自詡對 Debian 或 Linux 瞭若指掌。我所使用的安全性考量可能只適合於家庭使用。

本文件不能取代任何權威指南。

不提供任何保證。所有商標的所有權均屬於個別商標的所有人。

#### A.3 回饋

歡迎對本文件提出建議和補充。請以 email 寄給 debian-reference 套件或個別翻譯套件 的 Debian BTS system (http://bugs.debian.org/)。使用 reportbug 可以更容易的提供 一個精確的錯誤報告。你也可以用英文寄 email 至 <osamu\#at\#debian.org> 給 Osamu Aoki (http://people.debian.org/~osamu/) 或是用個別翻譯的語言寄給其翻譯者。

雖然我住在美國,不過我不是道地的英語使用者。歡迎任何有關文法上的指正。

最好的回饋方式是 SGML 版本的 patch,但是純文字版的 patch 也是歡迎的。請參閱官方文件 網站 '官方文件'1。

建立這份文件的原始 SGML 檔也可以 CVS 方式取得。CVS 位於: :pserver:anonymous@cvs.sf.net/cvsroot/qref或http://qref.sourceforge. net/Debian/qref.tar.gz.

### A.4 文件格式

這份文件是以 DebianDoc SGML DTD (由 LinuxDoc SGML 改寫而來) 來編寫。 DebianDoc SGML 系統讓一份原始文件可產生各種格式的文件,例如,本文件可以用 HTML,純文字, TeX DVI, PostScript, PDF 和 GNU info 等方式來閱讀。

DebianDoc SGML 的轉換工具可在 / debiandoc-sgml套件中取得。

### A.5 Debian 迷宮

Linux 系統是一個基於連網電腦的強大計算平台。然而,學習使用它所有的功能並非易事。設 定印表機就是個好例子。

有一張完整而詳盡的地圖叫做"SOURCE CODE"。它非常準確但極難理解。還有一些參考書叫 HOWTO 和 mini-HOWTO。它們易於理解但給出了太多細節反而讓人迷失了方向。爲了使用 某個命令,我有時得在長長的 HOWTO 中找上半天。

爲了在 Linux 系統設定的迷宮裡找到出路,我開始用文字檔的格式寫下簡單的備忘錄,作爲我 快速參考之用。當這些備忘錄越積越多之時我學會了 debaindoc 。於是 Debian 參考手冊 就產 生了。

## A.6 Debian 谚語

這裡有一些來自於 Debian 郵件列表的有趣谚語。

- "This is Unix. It gives you enough rope to hang yourself." —Miquel van Smoorenburg <miquels@cistron.nl>
- "Unix IS user friendly... It's just selective about who its friends are." —Tollef Fog Heen <tollef@add.no>